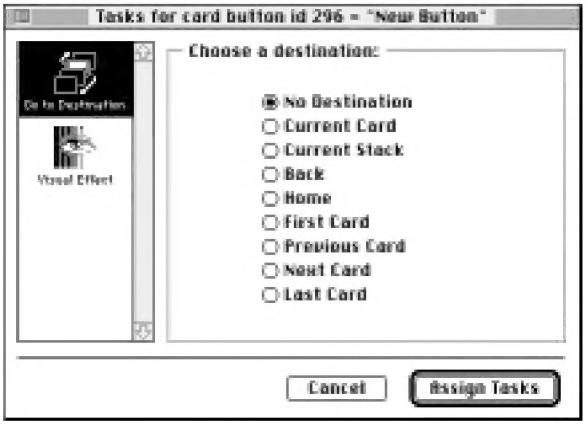
1 Using the Button Tasks Window

You use the Button Tasks window to add the power of scripts to HyperCard without the need to know how to write them.



Button Tasks is a Mac OS 7 feature: In order to use Button Tasks, you must be using Mac OS 7.0 or higher.

Opening the Button Tasks window

To open the Button Tasks window, follow these steps:

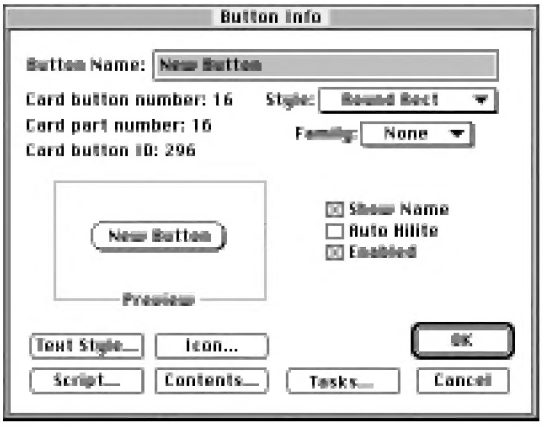
1. **Select the Button tool on the Tools palette.**

2. **Select any button.**

   If no buttons exist yet, choose New Button from the Objects menu to create one.

3. **Choose Button Info from the Objects menu.**

   A dialog box appears.



4. **Click Tasks.**

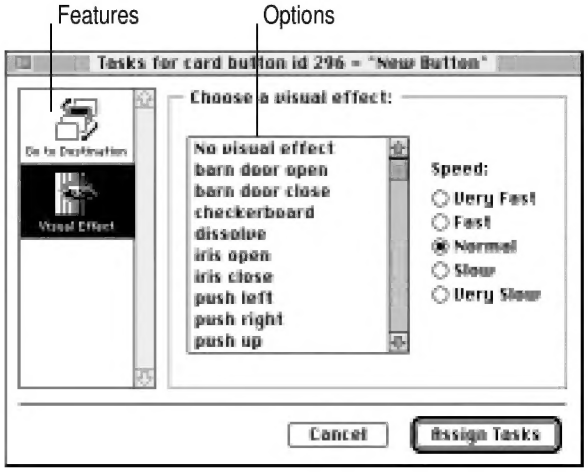
   The Button Tasks window appears.

Choosing tasks

To assign a task to the button, follow these steps:

1 **Click a feature on the left side of the Button Tasks window.**

The right side of the window displays options for that feature.



2 **Click the option or options you want to use.**

Some features let you set more than one option.

3 **Repeat steps 1 and 2 to assign additional tasks to the button.**

4 **When you've finished, click Assign Tasks.**

HyperCard creates a script for the button that adds the features you want and closes both the Button Tasks window and the Info dialog box.

To throw away changes and to close the Button Tasks window, click Cancel.

To see the complete script for the button, click Script in the Info dialog box. (When you've finished viewing the script, click the close box to put the script away.)

Assigning built-in tasks

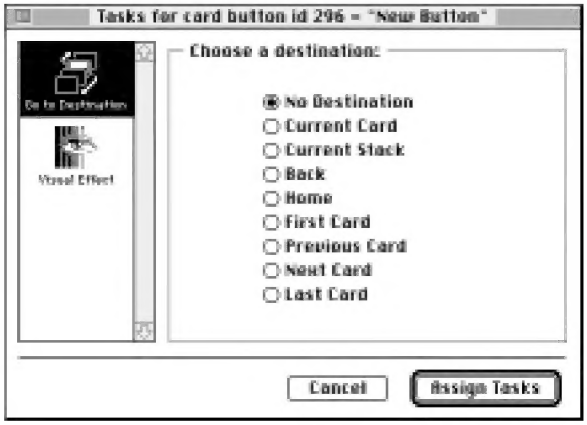
HyperCard 2.3 comes with several built-in tasks. You may see additional features in the Button Tasks window. You assign them in the same way you assign the tasks described below.

Going to a card

You can set a button to go to a card in this or another stack. Follow these steps:

1. **Click Go to a Card on the left side of the Button Tasks window.**

   The right side of the window displays a list of destinations.



2. **If the card you want to move to is in another stack or in a location that isn't listed, move to that card.**

   You may have to open another stack to locate the card.

3. **Click the description of the card you want to move to.**

The following table lists the tasks and their actions.

Task	Action
No Destination	removes any task already assigned that takes you to a card
Current Card	active card currently on the screen
Current Stack	first card of currently active stack
Back	card you were on when you moved to the current card
Home	first card in the Home stack
First Card	first card in the current stack
Previous Card	card before the current card in the current stack
Next Card	card after the current card in the current stack
Last Card	last card in the current stack

4 **Click Assign Tasks.**

Or click another icon on the left side of the dialog box and continue adding tasks to the button.

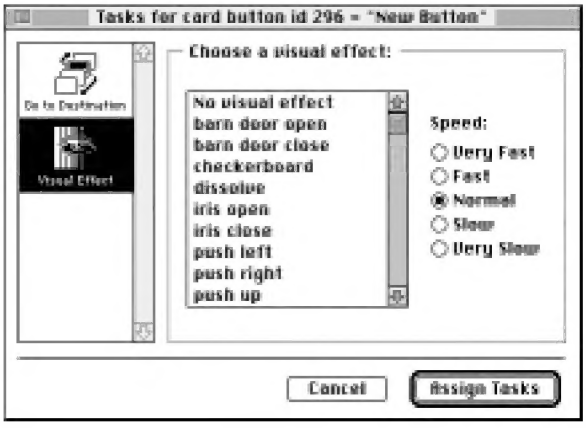
Adding a visual effect

You can assign a visual effect to a button to make movement between cards and stacks more noticeable and visually interesting.

If you're using color: If you'll be leaving or going to a card with color information on it, don't use a visual effect. Instead, use a transition effect in the color editor. See Chapter 4 for information.

1 **Click Visual Effect on the left side of the Button Tasks window.**

The right side of the window displays a scrolling field with an assortment of visual effects.



2 **Click the effect you want to use.**

3 **Click a speed.**

Speed determines how quickly the effect appears and disappears.

4 **Click Assign Tasks.**

Or click another icon on the left side of the dialog box and continue adding tasks to the button.

To test the new visual effect, choose the Browse tool from the Tools menu and click the button. You see the visual effect as you go to the card that the button is linked to.

Other features may be available: The tasks in the Button Tasks window are extensible; software developers can supply you with new tasks to add to the window.

2 Using the New HyperTalk Commands

HyperTalk for HyperCard 2.3 has new vocabulary for translating text to speech, delivering sound over several sound channels, and using the text contents of the clipboard as a container.

This chapter describes the following new HyperTalk words:

- `speak` translates a phrase into speech.
- `the speech` returns the text passed to the `speak` command.
- `stop speech` aborts the current and pending `speak` commands.
- `the voices` returns the voices available for generating speech.
- `the soundChannel` determines the channel on which the next sound will be generated.
- `stop sound` aborts all sound on all channels.
- `the clipboard` is a container that holds the textual contents of the clipboard.

For information on changes to existing HyperTalk commands, see Appendix B, "Updates." For information on HyperTalk commands that control color, see Chapter 9, "Scripting for Color."

Text to Speech

To use speech with HyperCard, you must have PlainTalk installed on your Macintosh. You need the Speech Manager extension, the MacinTalk Pro extension, and the Voices folder.

If you are using Mac OS 7.5 with a Macintosh AV computer or with a Power Macintosh computer, use PlainTalk version 1.3. Otherwise, use PlainTalk 1.2.1.

Speech sounds best on faster computers, such as the Macintosh Quadra models. On certain models with lower processor speed, such as the Macintosh II, some voices may be distorted.

IMPORTANT You cannot use PlainTalk version 1.2.1 on computers based on the 68000 central processing unit. Such computers include the Macintosh Classic and the Macintosh SE.

speak

```
speak phrase [with {genderLiteral voice | voice voice}]
```

```
speak "Hello, World."

speak field 3 with voice "Otis"

speak theSentence with female voice
```

phrase is any container or quoted string. *genderLiteral* is `male`, `female`, or `neuter`. *voice* is any voice in the Voices folder or in the MacinTalk file in the Extensions folder.

The `speak` command converts the text in *phrase* to speech, optionally using *voice*, and plays it through the device selected in the Sound control panel.

Speech is generated asynchronously and can be generated while HyperCard is in the background.

the speech

```
the speech
speech()
```

```
put the speech into currentSpokenPhrase
if speech() is "done" then speak "That's all, folks"
```

The `speech` function returns the currently generated speech text. If there is no currently generated speech text, speech returns `done`.

stop speech

```
stop speech
```

```
stop speech
if the time > "10:00 PM" then stop speech
```

The `stop speech` command stops the current speech and aborts pending speech commands.

To stop the current and all pending speech, as well as all current and pending sound, press ⌘-period.

the voices

```
the voices
voices()
```

```
put the number of lines in the voices into voiceCount
speak "Who's there?" with voice (any line of the voices)
```

The function `the voices` returns a return-delimited list of the voices currently available for generating speech.

For this function to work properly, all voices must either be in the Voices folder or in the MacinTalk file in the Extensions folder.

Sound

HyperCard supports up to eight channels for playing sounds. Channels can be played simultaneously or singularly.

IMPORTANT To use this feature, you need Mac OS 6.0.7 or higher operating on a computer with a 68020 or higher microprocessor. Some computers with 68020 or 68030 microprocessors may use fewer than eight sound channels.

the soundChannel

```
set [the] soundChannel to integerValue
```

```
set soundChannel to value(the soundChannel) + 1

if the soundChannel = 1 then play theTune
```

integerValue resolves to a whole number in the range 1 through 8.

The `soundChannel` property is the channel through which sound is played. The sound must have been generated by the `play` command.

The `play` command operates on the current sound channel. By immediately switching channels and playing new sounds, several sounds can be played nearly simultaneously.

```
on chord
  play harpsichord C E G
  wait 2 seconds
  set soundChannel to 1
  play harpsichord C
  set soundChannel to 2
  play harpsichord E
  set soundChannel to 3
  play harpsichord G
end chord
```

stop sound

```
stop sound
```

```
stop sound

if the voice is "Manager" then stop sound
```

The `stop sound` command stops the current sound and aborts pending sound commands on all channels.

To stop the current and all pending sounds plus all current and pending speech text, press ⌘-period.

To stop the sound on the current sound channel only, use `play stop`.

Clipboard

the clipboard

```
get [the] clipboard

put textValue into the clipboard
```

```
if the clipboard contains "Fred" then doMenu "Paste Text"

put the date && the time into the clipboard
```

`clipboard`, a new container, reflects the text contents of the clipboard.

IMPORTANT The expression `the clipboard` is guaranteed to give satisfactory results only when HyperCard is the frontmost application. To transfer information between applications, use AppleScript variables.

3 Installing Color Tools

This chapter describes how to prepare your Home stack so that you can use the color tools, how to turn the color tools on and off, and how to add color tools to a specific stack.

IMPORTANT To work effectively with color tools, HyperCard's preferred memory size should be set to 2200K or higher (the higher the better). The preferred memory size of HyperCard for Power Macintosh should be set to 5120K or higher. See the documentation that came with your computer for instructions on setting memory size.

Installing color tools into your Home stack

To use the color tools, you have to install them into your Home stack.

1 **Be sure that the Color Tools stack is in the same folder as the HyperCard application.**

2 **Open the Color Tools stack.**

Click the Color Tools button on the Stack Kit card of the Home stack, or double-click the Color Tools stack icon in the HyperCard folder.

The Color Tools screen appears.

3 **Click Install Color Tools.**

4 **When the Install card appears, click Install.**

(If you can't see the Install button, but instead you see a button labeled "Remove," color tools are already installed.)

During the installation process, a small text field appears over the Install button telling you what's going on.

When the installation is complete, the Install button changes its name to "Remove." If you want to remove the color tools from the Home stack, you can return to this stack and click Remove.

What Install does: Clicking Install makes changes to the Home stack script, adds the Color menu to the menu bar, and installs a new button on the first card of the Home stack.

Color Tools are ON

You can move the "Color Tools are ON" button to a different place on the Home card if you like.

Turning Color Tools on and off

When you first install the color tools, HyperCard adds the Color Tools stack to its "stacks in use" list. This allows other stacks to take advantage of the scripts and resources in the Color Tools stack.

You may want to shut off Color Tools because of some incompatibility problem. Clicking "Color Tools are ON" shuts off color tools—the Color menu disappears from the menu bar, the button changes to "Color Tools are OFF," and stacks no longer have access to the Color Tools scripts and resources.

Color Tools are OFF

Shutting off Color Tools doesn't shut off color: Turning off Color Tools won't remove color already displayed in a stack. Such color is permanently part of the stack.

Preparing a stack for color

To give a stack color capabilities, follow these steps:

1. **Go to the stack to which you want to add color.**
2. **Make a copy of the stack using the Save a Copy command from the File menu.**

   Conflicts occasionally occur between the resources in a stack and the resources that Color Tools add. Keeping an original copy of the stack makes it easy to recover from any problems.
3. **Choose Open Coloring Tools from the Color menu.**

   A dialog box appears, asking you to confirm that you want to add scripts and resources to your stack.

Opening the Color Editor will install scripts and resources in your stack to allow it to display color objects.
OK
Cancel

4. **Click OK.**

   When the installation is complete, the Color Editor opens.

   The color editor is described in Chapter 4; the color paint tools are covered in Chapter 7.

   Installing Color Tools increases the size of your stack by 34K and introduces free space into the stack. You can reduce the stack's size by choosing Compact Stack from the File menu.

Preparing a color standalone application

If you have added color resources to a stack, you need to close and reopen it before you save the stack as a standalone application. Closing and reopening the stack completes the installation of Color Tools.

Once the stack is saved as a standalone application, its memory allocation is the same as your version of HyperCard. You may have allocated extra memory to HyperCard that your application doesn't need. To lower the memory allocated to the application, follow these instructions:

1 **Make sure that the standalone application is not running.**

You can't change the memory allocated to a running application.

2 **Locate the standalone application.**

My Color Standalone

3 **Click the application's icon once to select it.**

4 **Choose Get Info from the File menu.**

You can also press ⌘-I.

A dialog box appears.

5 **Set the minimum memory size and the preferred memory size to 2200K (5120K for applications that may be used on Power Macintosh computers).**

6 **Click the Close box.**

Understanding the Color Editor

You use the color editor to add or change colors in a stack and to manipulate color PICT resources. This chapter presents a brief overview of the editor. Topics include

- color editor menus and commands
- color transition effects
- the color palette
- the basics of the RGB (Red-Green-Blue), HSB (Hue-Saturation-Brightness), and HSL (Hue-Saturation-Lightness) color mixing systems

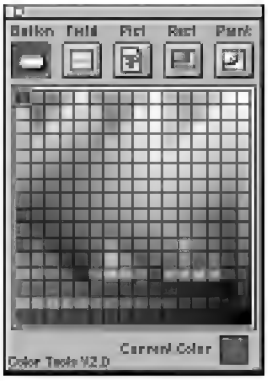
See Chapter 5 for information on colorizing buttons and fields, Chapter 6 for information on using PICT resources and files, and Chapter 7 for information on creating PICT images.

Create a practice stack: To learn the most, create a practice stack with several fields and buttons in both the card and background with which to experiment as you go through this and the next several chapters. See Chapter 3, Installing Color Tools, for instructions on adding color tools to your practice stack.

- **To open the color editor, choose Open Coloring Tools from the Color menu.**

  If this is the first time you're opening the editor in this stack, a dialog box appears asking you to confirm that you want to add color tools to the stack. Click OK.

  Any open palettes disappear, and the color editor opens. Color editor menus and the color palette appear.



  Closing the palette exits the editor: Clicking the close box on the color palette closes the color editor and returns you to the standard HyperCard environment.

Understanding overlays and layers

When you're in the color editor, you create or manipulate color overlays. A *color overlay* is any of the following: a PICT image displayed in the stack by using a color editor command, a color rectangle, or an element that adds the appearance of color to a button or field.

All color exists in a layer behind all paint pictures, buttons, and fields. The color layer holds all the color elements—button overlays, field overlays, rectangles, and PICT images. Color looks as if it's on top of other elements because the Color Tools make all white areas transparent (including opaque fields), allowing the color to show through. You can think of color as a piece of tinted cellophane laid behind a transparent object and cut to fit it.

Here's the order in which all HyperCard elements appear, from farthest back to topmost:

1. Color layer
   A. Background color overlays and PICT images, lowest to highest index number
   B. Card color overlays and PICT images, lowest to highest index number
2. HyperCard's background layer
   A. Background picture
   B. Background buttons and fields, lowest to highest part number
3. HyperCard's card layer
   A. Card picture
   B. Card buttons and fields, lowest to highest part number

Using color editor menus

When you open the color editor, you enter a new environment with a new set of menus and commands. You've left most of the familiar HyperCard world behind.

The following tables list the commands and describe their results.

File menu

Command	Action
Quit HyperCard	closes the color editor and quits HyperCard

Edit menu

Command	Action
Undo	undoes a color change, a size change to a PICT image, or a move of any kind
Cut	removes a color overlay (but not the object with which the overlay is associated)
Compact Color Database	eliminates color information about deleted objects
Sort Color Database	associates color overlays with their objects; arranges colorized buttons and fields in front of colorized rectangles and PICT images
Background	opens the background for editing or adding color overlays

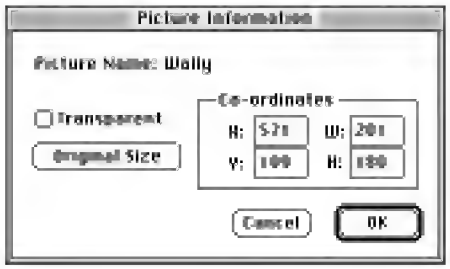
Items menu

Command	Action
Bring To Front	moves the selected color overlay (but not its object) in front of other color overlays; analogous to ⌘–Shift– + (plus) for a selected button or field
Bring Closer	moves the selected color overlay (but not its object) one layer closer to the front; analogous to the Object menu's Bring Closer command for a selected button or field
Send Farther	moves the selected color overlay (but not its object) one layer back; analogous to the Object menu's Send Farther command for a selected button or field
Send To Back	moves the selected color overlay (but not its object) behind other color overlays; analogous to ⌘–Shift– - (minus) for a selected button or field
Place Picture	displays a PICT image stored as a resource in this or any stack; retrieves and displays a PICT image from a file
Create New Picture	enters the PICT editor so you can create a new color PICT image
Edit Picture	if a PICT image is selected, opens the PICT editor and displays the selected PICT image in an editing window; otherwise, opens the PICT editor and displays an empty editing window
Place Rectangle	creates a new rectangle in the center of the screen using the currently selected color
Item Info	shows information about the selected color overlay; presents editing options (see "Using the Item Info Dialog Box," below)

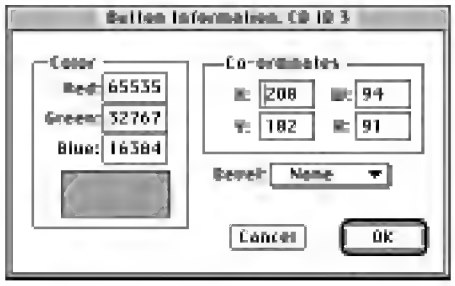
Using the Item Info dialog box

You use the Item Info dialog box to see information about the selected color overlay and to perform some editing functions.

If you've selected a PICT image, the dialog box displays the name of the PICT image and allows you to make its white areas transparent and to change its location or size. For more information, see Chapter 6, "Using Pictures," and Chapter 7, "Working with Color Paint Tools."

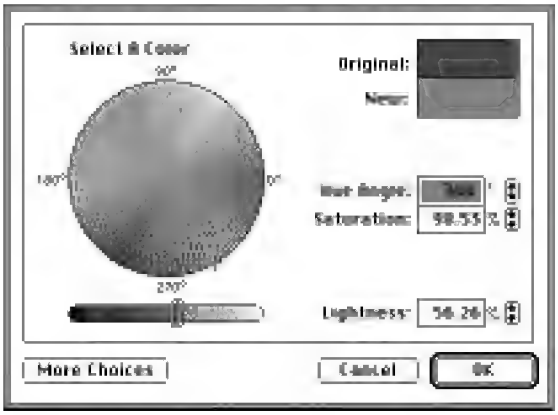


You use the Info box for a colorized button or field to change the element's color, location, or size, or to create a 3-D effect.



- **To open the color picker, click the color sample in a color overlay's Info box.**

  The color picker in Mac OS 7.1 is slightly different from the one in Mac OS 7.5. They work in essentially the same way.



- **To select a color, click the color you want to use.**

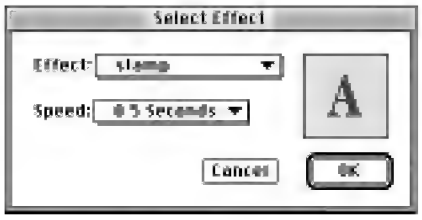
  The new color appears in the bottom half of the color sample.

  To restore the original color, click the top half of the color sample.

  When the color you want is in the bottom half of the color picker's color sample, click OK.

Effects menu

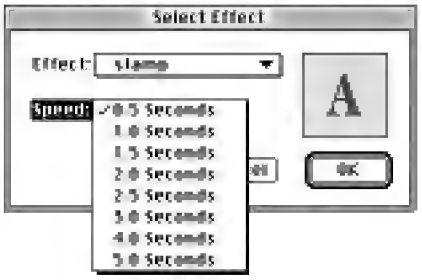
Command	Action
Stack Transitions	sets the default transition effect and duration for any card that doesn't already have a transition effect
Background Transitions	sets the default transition effect and duration for the current background; background transitions take precedence over stack transitions
Card Transitions	sets the transition effect and duration for the current card; takes precedence over all other transition settings



As you make settings, the sample window shows what the effect will look like.

Note: The transition effect dialog box does not indicate the name of the current effect.

The speed setting is relative to the machine you are using. The effect may take longer than that shown in the sample window. Card changes take longer in colorized stacks than they do in black and white stacks.



Why standard visual effects don't work

Commands in the Effects menu determine the transition effect in colorized stacks. The color editor adds a `closeCard` HyperTalk handler with a `lock screen` command to the stack. This command prevents HyperCard's standard visual effects from working as you change cards.

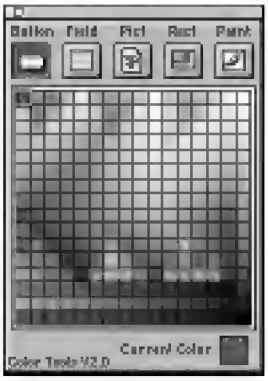
You can remove the `lock screen` command, but since HyperCard's visual effects work only on black-and-white images, if you follow a visual effect with any color effect, the image stutters as color is turned on.

Color menu

Command	Action
Open Coloring Tools	opens the color editor; on first use, modifies the current stack script and adds color resources to the stack
Redraw Screen	updates the display and realigns color overlays with the objects to which they belong (especially useful when you resize or move colorized buttons or fields)

Using the color palette

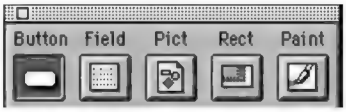
The color palette appears when you open the color editor. You use the color palette to choose a color for the currently selected shape, button, or field. It also provides shortcuts for certain menu commands.



Changing an element's color

To change the color of an overlay (any button, field, or shape), follow these steps:

1. **Click an icon at the top of palette.**



2. **Click the overlay you want to change.**
3. **Click a color on the color palette.**

The selected overlay changes to the color you've clicked.

You can set a PICT image to be transparent. If you want to change its color, use the PICT editor (described in Chapter 7).

The colors on the palette can't be changed. If you want a different color, you can use the color picker. (See the next section, "Coloring by the Numbers.")

Color palette icons

The icons at the top of the color palette enable different sets of tools and execute menu commands. The following table lists the effects of clicking each icon.

Icon	Effect
Button	enables selection of button overlays
Field	enables selection of field overlays
Pict	enables selection of PICT images; double-clicking initiates process of displaying a PICT image (a shortcut for the Items menu's Place Picture command)
Rect	enables selection of colorized rectangles; double-clicking creates a new colorized rectangle in the current color (a shortcut for the Items menu's Place Rectangle command)
Paint	opens PICT editor to edit currently selected PICT image (a shortcut for the Item menu's Edit Picture command) or to create a new one (a shortcut for the Items menu's Create New Picture command)

Coloring by the numbers

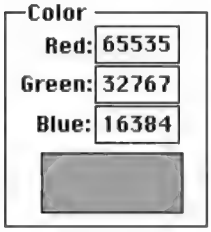
HyperCard provides a variety of ways for you to choose the color for your overlay.

- color palette
- color wheel
- RGB (Red-Green-Blue) color number system
- HSB (Hue-Saturation-Brightness) color number system (Mac OS 7.1)
- Apple RGB color number system (Mac OS 7.5)
- Apple HSL (Hue-Saturation-Lightness) color number system (Mac OS 7.5)

The number systems insure that an assigned color will appear exactly the same on computers that can show precise color differences. You don't have to know anything about the numbering systems to use colors in HyperCard.

RGB (Red-Green-Blue) system

The Info dialog box for colorized elements shows fields labeled Red, Green, and Blue—the elements of the RGB color system. The RGB system is based on the way that monitors display color. A color monitor has three electron streams—one each for red, green, and blue, the primary colors of light. Each pixel is given a certain intensity from each stream. Full intensity is represented digitally by the number 65535, and no intensity—the stream turned off—is represented by the number 0.



The color of a pixel depends on the mixture of intensities of the three streams. When you increase the intensities, the color gets brighter. If you mix all three at maximum intensity (65535, 65535, 65535), you get white. (Contrast this with house paint colors. When you mix primary colors together, the resulting color gets darker.)

For example, the RGB number set 65535, 0, and 0 produces pure red. As you increase the numbers for green and blue, the red changes to lighter shades and finally to white.

Mixtures of equal intensities from all three streams result in shades of gray. The closer the intensity is to 0, the closer the shade is to black.

- **To produce dark gray, use 20000, 20000, 20000.**
- **To produce light gray, use 50000, 50000, 50000.**
- **To get primary red, green, or blue, set that color to a number greater than zero and set both of the other numbers to zero.**
- **To make the color brighter, increase its number.**
- **To make the color duller, decrease its number.**
- **To make a pastel, add the other colors in equal amounts.**

  Pastel green is 40000, 65535, 40000.

- **To get secondary colors, set one of the colors to 0.**

  Red and green make yellow; green and blue make cyan; red and blue make magenta.

  For the most intense yellow, set red and green to 65535. To make it duller, turn down the intensity of both colors. To make it paler, add some blue. (A pale yellow is 65535, 65535, 30000.)

HSB (Mac OS 7.1)

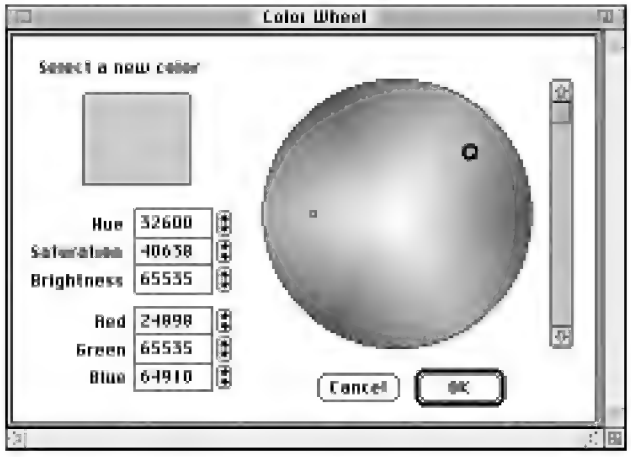
Mac OS 7.1 lets you control both RGB and HSB values by number. HSB (Hue-Saturation-Brightness) is an alternate way to describe colors, also based on three sets of numbers in the range 0 through 65535.

Hue defines the color, with 0 for red, 21845 for green, and 43690 for blue. (The numbers wrap around, so 65535 and 0 are the same red.)

Saturation is a measure of the amount of white in the color—65535 means no white is in the color (the color is pure); 0 means that there's no hue left; only white. A color with a high saturation is called hot, while one with low saturation looks pale.

Brightness describes how much black is in the color—65535 means no black is in the color (the color is pure); 0 means that there's no hue left and the color is black.

Changing the hue is equivalent to traveling around the color picker without changing the distance from the center. Changing the saturation is equivalent to traveling inward or outward along a spoke of the wheel. Changing the brightness is equivalent to moving the scroll bar to the right of the color picker.



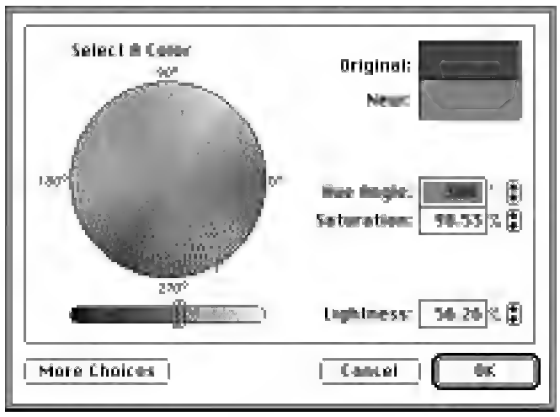
Apple HSL (Mac OS 7.5)

The color picker in Mac OS 7.5 shows Apple HSL (Hue-Saturation-Lightness) values.

The H value, Hue, is a number between 1 and 360 representing an angle—90° is yellow, 180° is blue, 270° is violet, and 360° is red.

The S value, Saturation, is a percentage. The higher the number, the greater the intensity of the color. For example, red with a saturation of 75% is a medium red; 100% saturation is bright red.

The L value, Lightness, is the percentage of white in the color—the less white, the darker the color; the more white, the paler the color. (Pastels have a greater percentage of white than pure colors.)



Apple RGB (Mac OS 7.5)

Apple RGB works like the RGB system described earlier in this section, except that you set values by percentage (0 through 100) rather than by number (0 through 65535).

The percentages correspond to the numbers. For example, an RGB set of 65535, 32767, 16384 corresponds to an Apple RGB set of 100, 50, 25 (a medium orange).

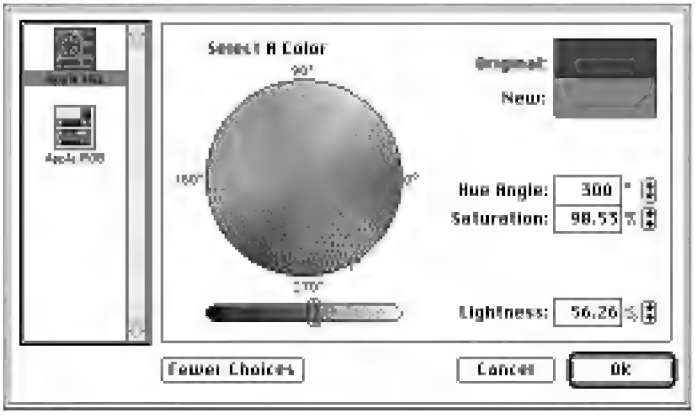
To use sliders to set RGB values, follow these instructions:

1. **Select an overlay in the color editor; then choose Item Info from the Items menu.**
2. **Click the color sample in the Info dialog box.**

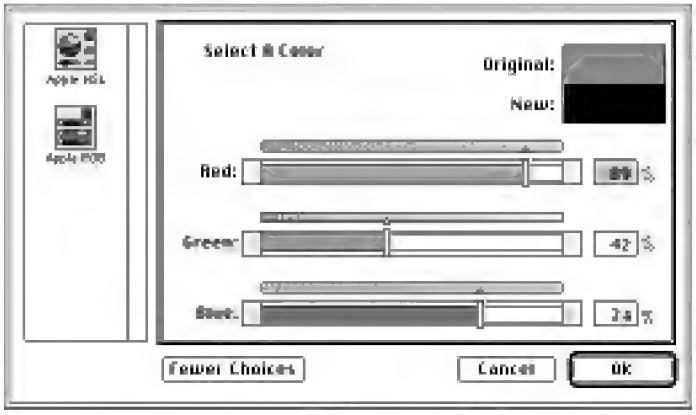
The color picker appears.

3 **Click More Choices at the bottom of the color picker dialog box.**

The dialog box expands.



4 **Click the icon labeled Apple RGB.**



5 **Drag a handle or click a bar to change the color balance.**

The new color is reflected in the color sample labeled "New."

To return to the original color, click Original.

5 Colorizing Buttons and Fields

You colorize buttons and fields by adding overlays to them in the color editor. You can also add 3-D effects to the overlay.

This chapter describes how to

- add color overlays to buttons and fields
- use the Info box to add 3-D effects, to modify a color, and to change an object's size or position
- use editing commands to modify a color overlay

When you edit a color overlay, the object itself remains unaffected. You can resize and move an object, but in order to make any other changes, you must quit the color editor.

Colorizing is only for buttons and fields: To add color to a card or background, you can draw a large rectangle and move it all the way to the back. See Chapter 4 for information.

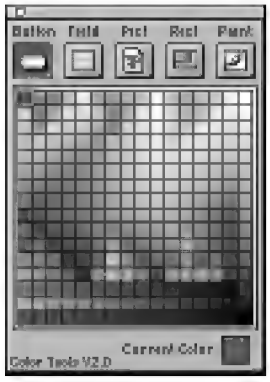
Adding color to an object

An object must already exist in the proper layer (card or background) before you can colorize it in the editor.

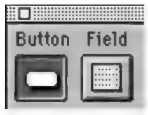
To add a color overlay to an object, follow these instructions:

1. **Create an object on the card or background layer.**
2. **Choose Open Coloring Tools from the Color menu.**

   The color editor opens.



3. **Click the button or field icon at the top of the color palette.**



4. **Click the object you want to colorize.**

   You can colorize only one object at a time.

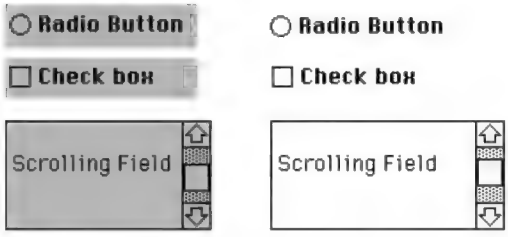
   Clicking an object selects it. If you click the button icon, you can select only a button; if you click the field icon, you can select only a field.

   How to find invisible objects: To find borderless uncolored buttons in the color editor, press Shift-Option. To find borderless uncolored fields, press ⌘-Shift-Option.

5 **Click a color on the Color Tools palette.**

If the palette doesn't include the color you want, you can use the color picker or one of the color numbering systems described in "Coloring by the Numbers" in Chapter 4. (To get to the color picker, click the current color at the bottom of the palette.)

The selected object changes to the color you've clicked. To avoid obscuring text, use light colors for fields and for buttons whose names show.



Colorized Objects Uncolorized Objects

All HyperCard buttons and fields become transparent as soon as you add color resources to the stack. If you want an opaque object, you can color it white.

Using the Info box

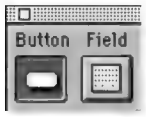
You can use a colorized object's Info box to

- adjust the color
- open the color picker
- set or change a 3-D effect
- change the position of the object and overlay
- change the size of the object and overlay

You can also control all of these settings in HyperTalk. See Chapter 9 for details.

To open an object's Info box, follow these instructions:

1. **If the color editor isn't already open, choose Open Coloring Tools from the Color menu.**
2. **Click the button or field icon, as appropriate, at the top of the color palette.**

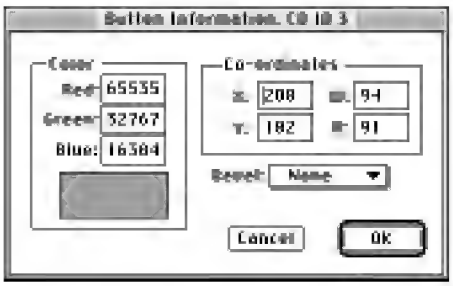


3. **Click the colorized object you want to work with and choose Item Info from the Items menu.**

Or you can double-click the object.

You must colorize an object before you can use its Info window.

The Info box opens, with the ID of the object in the title.



Adjusting RGB numbers

You can adjust an object's color by changing the RGB numbers in the Info box.

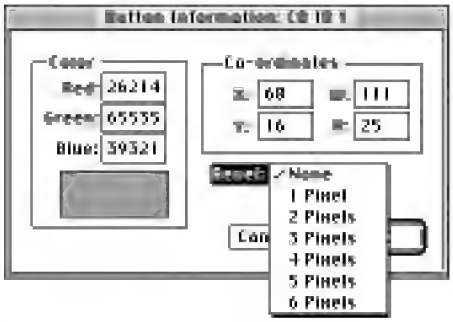
- **To modify an RGB number, select it and type the new number.**

  You can immediately see the change in the color sample. However, some changes in RGB numbers are too subtle to see, especially where you're displaying 256 colors or less. For more apparent changes, use the two leftmost digits.

  For detailed information about RGB numbers, see "RGB (Red-Green-Blue) System" in Chapter 4.

Adding 3-D effects

You use the Bevel pop-up menu to add shading to a selected object.



You can add bevels of 1 to 6 pixels—the higher the number, the wider the bevel and the heavier the shadowing.

Transparent objects don't show bevels. To shade a transparent object, color it white first.

Bevel tips

- Heavy bevels can obscure text, especially in fields. To avoid the problem, turn on the Wide Margins option in the field's Info dialog box or choose a lower bevel number.
- Avoid bevels on colorized checkboxes and radio buttons.
- Oval buttons with a 3-pixel bevel look great with pastel colors.

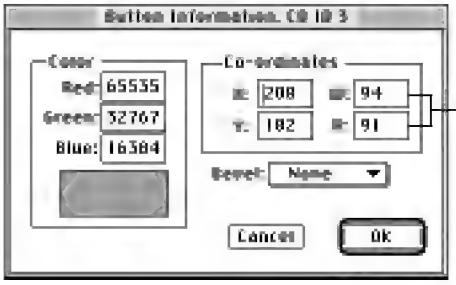
Changing an object's position

The easiest way to change an object's position is to drag it. You can position it more precisely, however, by setting its coordinates in the Info box's Co-ordinates window.

The X and Y coordinates represent a point X pixels to the right of and Y pixels down from the upper-left corner of the stack window. When you close the Info box, the upper-left corner of the object moves to the specified coordinates.

Changing an object's size

The easiest way to change a selected object's size is to drag its corner. You can precisely set its width and height in the Info box.



- To keep the object's proportions constant, change both numbers by the same percentage.

Editing a colorized object's overlay

You edit an overlay the same way you created it. Follow these instructions:

1. **If the editor isn't already open, choose Open Coloring Tools from the Color menu.**
2. **Click the button or field icon at the top of the color palette.**
3. **Select the object whose overlay you want to edit.**
4. **Make changes as appropriate.**

   Choose any menu item or select a new color on the color palette.

Using menu commands

In the color editor, the Undo command in the Edit menu undoes the most recent move, resizing, or color change for the currently selected colorized object. There is no Redo command.

The Cut command (or the Delete key) removes the color from the currently selected object. It does not remove the object itself.

Copying a color from one object to another

The Copy command places the color and bevel of the selected overlay onto the clipboard. You can use the Paste command to assign that color and bevel to any other object of the same type (either button or field). You cannot copy the color of one kind of object and paste it to another.

You can also write scripts to change colors. See Chapter 9, "Scripting for Color."

Copying an object does not copy its color: When you copy and paste (or drag-copy) a colorized button or field in HyperCard, the object's color is not copied. When you copy and paste or duplicate a card and paste it, the colors associated with that card aren't pasted onto the new card.

6 Using Pictures

This chapter describes how to display high resolution PICT files and resources, properly scaled and cropped, anywhere on a card or background. It also covers the effects of storing PICT images as resources in a stack and as files on disk.

For information on how to create and edit PICT images, see Chapter 7, "Working with Color Paint Tools."

You can use scripts to display PICT images: HyperCard includes several PICT scripting commands. See the sections on the `addPict`, `colorPict`, `addPictFile`, and `colorPictFile` commands in Chapter 9 for details. These commands complement the existing Picture command, described in the *HyperCard Script Language Guide*.

PICT resources versus PICT files on disk

PICT resources reside in the resource fork of the stack in which they are displayed. PICT files are stored on a CD ROM disc, floppy disk, or hard disk.

PICT resources

PICT resources move with the stack when you copy or move it.

PICT resources increase a stack's size. (Stacks with several PICT resources often outgrow high-density floppy disks.)

PICT files on disk

PICT files are easy to transport. You can save space by using a compression utility.

You must remember to keep all the PICT files with the stack when you move it.

The PICT files must go in the same folder as HyperCard, Home, or your stack. (The color editor cannot store the entire path name of a PICT file.)

The Color Tools can display only the System Palette colors. If your PICT image was created using different colors, the Color Tools automatically convert those colors to the system colors.

Adding a PICT image to your stack as a resource

You can copy a PICT file's image and store it as a PICT resource in your stack, or you can copy a PICT resource from another stack.

Adding a PICT file from disk

1. **Choose Open Coloring Tools from the Color menu.**
2. **Choose Place Picture from the Items menu.**

Or double-click the Pict icon at the top of the color palette.

A dialog box appears with a list of the PICT images already installed in the stack.

3 **Click Import.**

A dialog box appears.

4 **Locate the PICT file you want to add and click Open.**

A copy of the PICT image is stored in your stack. (The original remains undisturbed on the disk.)

5 **When you've finished adding PICT images, click Cancel to close the dialog box and return to the color editor.**

Copying a PICT image from another stack

1 **Choose Open Coloring Tools from the Color menu.**

The color editor opens.

2 **Choose Place Picture from the Items menu.**

Or you can double-click the Pict icon at the top of the color palette.

A dialog box appears with a list of the PICT images already installed in the stack.

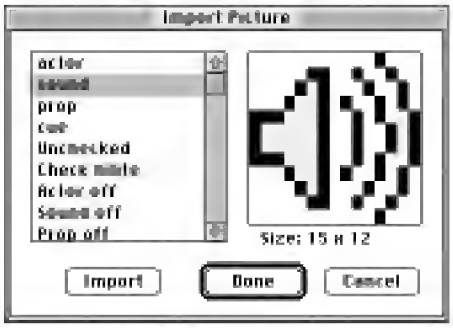
3 **Click Import.**

A dialog box appears.

4 **Locate the stack with the PICT image you want and click Open.**

A dialog box appears with a list of PICT images you can copy to your stack.

PICT images whose names start with a bullet (•) don't appear in the list and cannot be imported.



5 **Select the PICT image you're interested in.**

The PICT image appears in the preview window, with its size (in pixels) displayed below the window.

6 **Click Import to copy the PICT image to your stack.**

7 **When you've finished, click Done.**

You return to the Select a Picture dialog box. The name of the PICT image you've imported appears in the list.

8 **Click Cancel to close the dialog box and return to the color editor.**

Displaying a PICT resource

You display a PICT resource in your stack by making a choice from the Select a Picture dialog box.

1 **Be sure that you're in the layer (card or background) where you want the PICT image to appear.**

2 **With the color editor open, choose Place Picture from the Items menu.**

Or double-click the Pict icon from the top of the color palette. The Select a Picture dialog box appears with the names of all the PICT resources for the current stack.

3 **Click the name of the PICT image you want to show.**

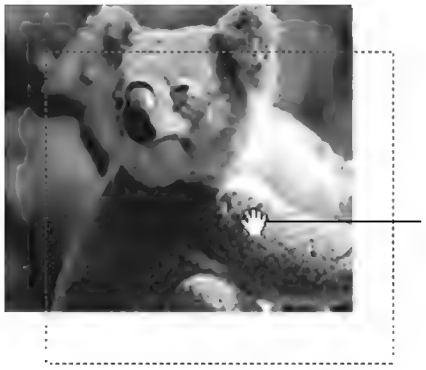
A miniature of the PICT image appears in the dialog box's preview window, with its size in pixels. (The image may appear distorted.)

4 **Click Place.**

The PICT image appears selected on the display.

5 **Drag the PICT image where you want it to be.**

Be sure to drag it by its middle. (Dragging a PICT image by its edges changes its proportions.) The pointer changes to a hand to indicate that you'll be moving the PICT image.



Displaying a PICT file on disk

To use a PICT file in your stack, follow these steps:

1 **Be sure that you're in the layer (card or background) where you want the PICT image to appear.**

You can't cut a PICT image from one layer and paste it into another.

2 **With the color editor open, choose Place Picture from the Items menu.**

Or double-click the Pict icon at the top of the color palette.

A dialog box appears.

3 **Click PICT File.**

Another dialog box appears.

4 **Find the PICT file you want to display.**

5 **Click Open.**

The PICT image appears selected on the display.

6 **Drag the PICT image where you want it to be.**

Be sure to drag it by its middle.

Picture files and search paths: PICT files or aliases for them must be in the same folder as the HyperCard program, the Home stack, or your stack. If you use an alias for a PICT file, its name must be the same as the name of the original file.

Scaling PICT images

Scaling a PICT image changes its height, width, or both. You can change a selected image's proportions by dragging one of its edges in the stack window.

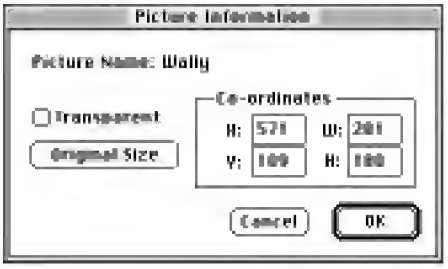
You can scale a PICT image more precisely within its Info box. You can also scale PICT images from within the PICT editor. See "Scale" in Chapter 7 for information.

1 **With the color editor open, click the Pict icon at the top of the color palette.**

2 **Click the PICT image you want to scale, and choose Get Info from the Items menu.**

Or double-click the PICT image.

The PICT image's Info box appears.



3 **Type new numbers into the Width (W) and/or Height (H) boxes.**

To keep the proportions constant, change the W and H numbers by the same percentage.

4 **Click OK.**

The image changes after you close the Info box.

- **To restore the image to its original size and proportions, click Original Size in the PICT image's Info box.**

  You can use this technique even if you've changed the PICT image's dimensions by dragging its corners.

Cropping PICT images

To crop a PICT image, you use the marquee tool from the color paint tools palette.

1 **Click a PICT image to select it.**

2 **With the color editor open, click the Paint icon at the top of the color palette.**

The PICT editor opens and the color paint tools palette appears.

3 **Using the marquee tool, select the area you want to crop.**

4 **Press Return to erase the selected area.**

You can use other tools for more sophisticated cropping. See Chapter 7 for details about using the color paint tools.

Dithering

Sometimes the PICT image you want to use has more colors than your computer system can display. For example, you might want to display a PICT image that was scanned using millions of colors, but your current monitor can show only 256 colors.

In such a case, your computer uses what it considers to be the closest available colors from the system palette. Sometimes this causes banding in large swatches of gradually changing color.

HyperTalk's Picture command overcomes these problems by using dithering, a process that helps the computer approximate colors. Dithering creates patterns made up of available colors to simulate missing colors. For details on the Picture command, see the *HyperCard Script Language Guide*.

Change the Color Tools' bit depth: If your monitor can display more than 256 colors, you can use HyperTalk to change the bit depth that the Color Tools uses. In the stack's openStack handler, replace the line `addColor install` with `addColor install, 16` or `addColor install, 24` (depending on whether your monitor can display thousands or millions of colors). Note that using a greater bit depth requires more memory and slows down stack execution.

7 Working with Color Paint Tools

This chapter explains how to use the color paint tools to create and edit color PICT images, and how to modify PICT images that you import from other stacks or from PICT files.

Entering the PICT editor

The color paint tools are part of the PICT editor. You open the PICT editor by clicking the Paint button on the color palette in the color editor.

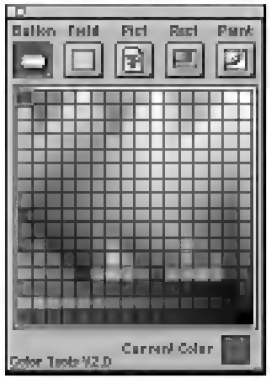
The images you create or edit in the PICT editor are automatically saved as resources in the current stack when you close the editor.

Preparing to edit an existing PICT image

To edit an existing PICT image, follow these steps:

1. **Choose Open Coloring Tools from the Color menu.**

The color editor opens with the color palette on the display.



2. **Click the Pict icon at the top of the color palette.**

3 **Click the PICT image you want to edit.**

Or double-click the Pict icon; then choose a PICT resource from the list that appears.

For detailed instructions on importing PICT resources or PICT file images, see Chapter 6.

4 **Choose Edit Picture from the Items menu.**

Or click the Paint icon at the top of the color palette.

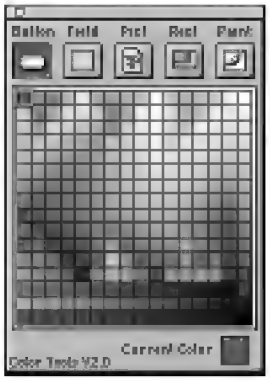
The PICT editor opens, and the color paint tools palette replaces the color palette. The PICT image you've selected appears in a new window.

Preparing to create a new PICT image

To create a new PICT image, follow these steps:

1. **Choose Open Coloring Tools from the Color menu.**

   The color editor opens with the color palette on the display.



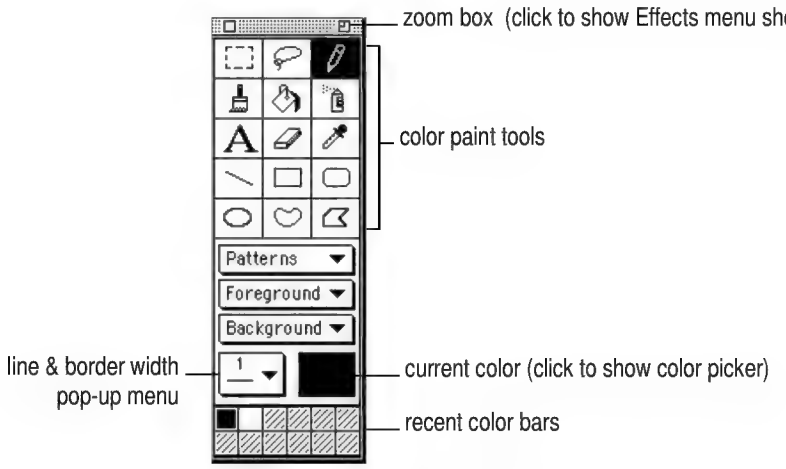
2. **Choose Create New Picture from the Items menu.**

   Or double-click the Paint icon at the top of the color palette.

   The PICT editor opens, and the color paint tools palette replaces the color palette. A new window appears in which you can draw a new PICT image.

Using the color paint tools palette

You use the color paint tools palette to create and edit PICT images.



To add color to an image, follow these steps:

1. **Click the PICT window whose image you want to edit.**
2. **Choose a pattern from the Patterns pop-up menu.**
3. **Choose a foreground color from the Foreground pop-up menu.**

   Or click a color on the Recent Color bars.

   The foreground color is for the part of the pattern that shows as black in the Patterns menu. Thin straight lines, thin borders around shapes, and plain text also appear in the foreground color.

4 **Choose a background color from the Background pop-up menu.**

Or Option-click a color on the Recent Color bars.

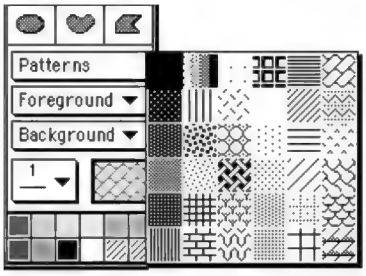
The background color is for the part of the pattern that shows as white in the Patterns menu. The inside parts of wide lines, wide borders, and hollow text also appear in the background color.

5 **Choose a color paint tool and edit the image.**

The color sample (called the *current color*) near the bottom of the palette is a composite of the current pattern, foreground color, and background color. Whatever you paint uses this composite.

Choosing and editing patterns

You use the Patterns pop-up menu to choose a pattern. The black pixels in a pattern will appear in the current foreground color; the white pixels will appear in the current background color.



- **To choose a pattern, drag to the one you want to use in the Patterns pop-up menu; then release the mouse button.**

  The pattern appears in the current color box.

- **To edit a pattern, choose Edit Patterns from the Options menu, click the pattern, and then edit it pixel by pixel.**

Using the color paint tools

- **To select a tool, click it.**

  The shape of the pointer may change depending on the tool you select.

  Most of the color paint tools work like HyperCard's black-and-white paint tools.

Marquee tool

Use the marquee tool to select a rectangular area of an image.

- **To select an area, drag diagonally across it.**
- **To select the entire image, double-click the marquee tool icon in the Paint palette.**

  Or choose Select All from the Edit menu.

- **To shrink to the selection, hold down the Option key as you drag.**

  When you release the mouse button, the selection rectangle shrinks around the image, eliminating the extra white space.

- **To move a selection, put the pointer inside a selection and press the mouse button until the pointer changes to an arrow; then drag.**
- **To move a selection straight up or down or left or right, press the Shift key as you drag.**
- **To make a copy of the selection, press the Option key as you drag.**
- **To make multiple copies of a selection, press ⌘–Option as you drag.**
- **To move an image one pixel at a time, select it; then press an arrow key.**
- **To move an image five pixels at a time, select it; then hold down the Shift key as you press an arrow key.**

Lasso tool

Use the lasso tool to select any of the following:

- a specific part of an image too small for the marquee tool
- a non-rectangular part of an image
- an image that is between other graphics

- **To use the lasso tool, drag around the area you want to select.**

  The line that trails the lasso's tip shows what you are selecting. When you release the mouse button, the lasso shrinks around the image and selects it.

- **To lasso an area without shrinking around the image, press the Option key as you drag.**
- **To lasso the whole image without selecting white space, double-click the lasso icon.**
- **To move a selection, position the lasso tool's tip within the selection until the pointer changes to an arrow; then drag.**
- **To move a selection straight up or down or directly left or right, press the Shift key as you drag.**
- **To make a copy of the selection, press the Option key as you drag.**
- **To make multiple copies of the selection, press ⌘–Option as you drag.**
- **To move an image one pixel at a time, select it; then press an arrow key.**
- **To move an image five pixels at a time, select it ; then hold down the Shift key as you press an arrow key.**

Pencil tool

Use the Pencil tool to draw thin free-form lines.

The pencil draws in the selected color. If you click a pixel of the selected color and then draw, the pencil draws in the background color.

- **To draw straight lines in a vertical or horizontal direction, press the Shift key as you drag.**
- **To zoom in to a higher magnification, double-click the pencil icon.**

  Or ⌘-click the area that you want magnified.
- **To zoom out to standard magnification, ⌘-Shift-click.**
- **To move the entire work area, press the Option key as you drag.**

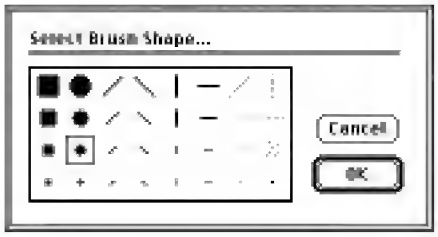
Brush tool

Use the brush tool to paint with the current pattern and brush shape.

- **To use a new brush shape, choose Select Brush from the Options menu.**

  Or double-click the brush in the tool palette.

When the dialog box appears, click the shape you want to use.



Using different brush shapes and patterns creates different effects.

- **To paint straight lines in a vertical or horizontal direction, press the Shift key as you drag.**
- **To use the brush tool as an eraser, press the ⌘ key as you drag.**

Paint bucket tool

Use the paint bucket tool to fill solid areas of an image or the hollow parts of outlined and shadowed text with the selected pattern and color.

IMPORTANT Zoom in to check for gaps before using the paint bucket tool. If there is a gap or space in the outline of an area, paint spills out and fills the surrounding area. (You can choose Undo from the Edit menu to correct the mistake.) See "Zooming" later in this chapter for instructions.

- **To fill an area with the currently selected pattern and color, place the tip of the bucket in the area and click.**

Spray can tool

Use the spray can tool to spray color onto an area.

- **To bring up a dialog box to set the width (Aperture) and density (Pressure) of the spray, double-click the spray can icon.**

A Text tool

Use the text tool to add text to the image.

You can change the text font, size, and style by making choices from the Text menu immediately after typing the text and before clicking anywhere on the screen.

- **To select the text you just typed, click the marquee or lasso tool.**
- **To select a font, choose Font from the Text menu and choose a font from the submenu.**

  The current font is checked.
- **To select a size, choose Size from the Text menu and choose a size from the submenu.**

  The current size is checked. The best sizes for the current font are outlined.
- **To see the Text Size dialog box, double-click the text tool icon.**
- **To select unlisted sizes, choose Size from the Text menu and drag to Other.**

  When the dialog box appears, click the up or down arrow until you see the size you want to use in the text box (or type the size into the text box).
- **To select a style for your text, choose Style from the Text menu and drag to the style you want to use.**

  You can choose multiple font styles for the same text.
- **To justify your text, choose Style from the Text menu and drag to the justification you want to use.**

Eraser tool

Use the eraser tool to erase part or all of the image.

- **To erase part of the image, drag over the part you want to remove.**
- **To erase the entire image, double-click the Eraser tool icon.**

  If you make a mistake, choose Undo from the Edit menu.

Color pick-up tool

Use the color pick-up tool to select a color from part of the image.

- **To pick up any color on the screen, position the color pick-up tool over the color and click.**

  The current color box shows a solid foreground pattern and the color you click.
- **To change all instances of a color to the current color, Option-click the color with the color pick-up tool.**
- **To change all instances of a color to the current gradient, ⌘-Option-click the color with the color pick-up tool.**

  See "Creating a Gradient" later in this chapter for information on gradients.
- **To toggle between the current tool and the color pick-up tool, press the Tab key.**

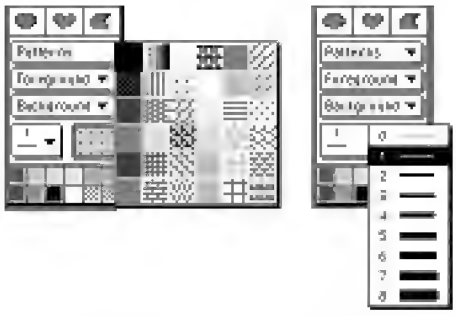
Line tool

Use the line tool to draw straight lines.

- **To draw lines in the selected pattern, press the Option key as you drag.**
- **To draw straight horizontal or vertical lines, press the Shift key as you drag.**
- **To change the line thickness, use the Line Width pop-up menu on the palette.**

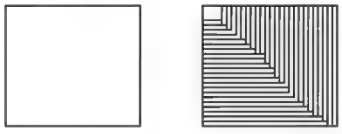
Creating closed shapes

- **To create any closed shape, click a shape tool; then select a pattern and a border size from the palette's pop-up menus.**



- **To draw centered on a given point, choose Draw Centered from the Options menu before you drag.**
- **To draw borderless shapes, choose Draw Filled from the Options menu and press the Option key as you drag.**

  Or select 0 from the Line Width pop-up menu.
- **To draw multiple shapes, choose Draw Multiple from the Options menu before you drag.**



Draw Multiple off Draw Multiple on

- **To create squares or circles, press the Shift key as you drag.**
- **To draw rectangles, use the rectangle tool.**

- **To draw rectangles with rounded corners, use the rounded rectangle tool.**

- **To draw oval shapes, use the oval tool.**

- **To draw a curved shape, position the pointer where you want to start; then drag.**

When you release the mouse button, the tool automatically draws a straight line connecting the start and end points of the shape.

Polygon tool

Use the Polygon tool to create polygons with irregular sides.

1. **Position the mouse where you want to begin; then click.**
2. **Without pressing the mouse button, move the mouse to a second point and click, then to the third point and click, and so on until you're finished.**
3. **To complete the polygon, double-click.**

You can now move the mouse without drawing any more lines.

Setting effects

The PICT editor provides a wide range of paint effects.

- **To apply an effect to an entire PICT image, choose the effect you want to use from the Paint or Options menu.**

  The effect will change the PICT image in the active PICT window.

- **To apply an effect to part of a PICT image, select the part you want to enhance; then choose an effect from the Paint or Options menu.**

  The effect will change only the selected part of the PICT image in the active PICT window.

The wording of a menu command may change to reflect whether that command will enhance the entire PICT image or just the part you have selected.

Fill

- **To fill the image or the selection with the current pattern and colors, choose Fill from the Effects menu.**

  Or use the bucket tool.

Invert

- **To change the color of pixels to colors on the opposite side of the color spectrum, choose Invert from the Effects menu.**

Tint

- **To tint the image by adding the foreground color to pixels, choose Tint in the Effects menu; then choose Toward Foreground in the submenu that appears.**
- **To tint the image by adding the background color to pixels, choose Tint in the Effects menu; then choose Toward Background in the submenu that appears.**

Anti-Alias

- **To remove jagged edges and outlines from the image, choose Anti-Alias from the Effects menu.**

  Anti-aliasing smooths the edges of the image by creating grayish intermediate pixels so that the edges become slightly blurred.

Trace Edges

- **To outline edges, choose Trace Edges from the Effects menu.**

  Or press ⌘-E.

  Repeated tracing adds more outlines to the edges.

Rotate

- **To turn the image 90 degrees right, choose Rotate in the Effects menu; then choose Right in the submenu that appears.**
- **To turn the image 90 degrees left, choose Rotate in the Effects menu; then choose Left in the submenu that appears.**
- **To turn a selected image by hand to any degree of rotation, choose Rotate in the Effects menu; then choose Free in the submenu that appears.**

  Drag the image by the handles that appear.

- **To turn the image by a specific number of degrees, choose Rotate in the Effects menu; then choose By Degree in the submenu that appears.**

  When a dialog box appears, type the number of degrees of rotation that you want, click Clockwise or Counter-clockwise, and click OK.

Scale

- **To scale a selected image by hand to any percentage, choose Scale in the Effects menu; then choose Free in the submenu that appears.**

  Drag the image by the handles that appear.

- **To scale the image by a specific percentage, choose Scale in the Effects menu; then choose By Percent in the submenu that appears.**

  When a dialog box appears, type the percent scaling that you want.

  To dither the image when it scales, click Use Dithering in the Scale dialog box. (See "Dithering" in Chapter 6 for information about dithering.)

  To scale the image proportionately, use the same number in both boxes.

Flip

- **To flip the image vertically about its center line, choose Flip in the Effects menu; then choose Vertically in the submenu that appears.**
- **To flip the image horizontally about its center line, choose Flip in the Effects menu; then choose Horizontally in the submenu that appears.**

Opaque

- **To make the image opaque so that nothing shows through white parts, choose Opaque from the Effects menu.**

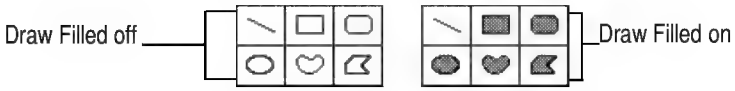
Transparent

- **To make the image transparent so that anything below white parts shows through, choose Transparent from the Effects menu.**

Draw Filled

- **To fill shapes with the current pattern and colors as you draw them, choose Draw Filled from the Options menu.**

  This command affects the shape tools. They appear filled when this command is in effect.



Draw Centered

- **To draw a shape starting at the shape's center point, choose Draw Centered from the Options menu.**

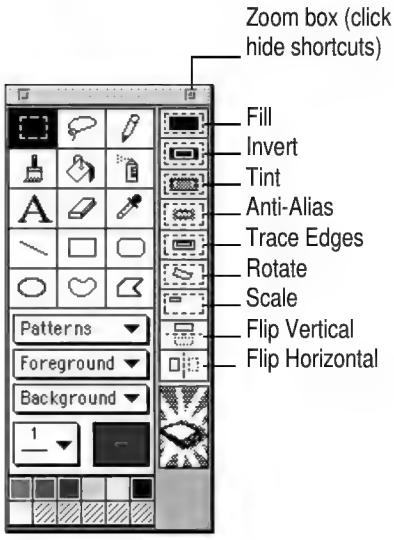
Draw Multiple

- **To draw multiple images as you drag a tool, choose Draw Multiple from the Options menu.**

  This command affects the Line, Rectangle, Rounded Rectangle, and Oval tools.

Paint effects shortcuts

When you click the Zoom box on the color paint tools palette, the palette expands to show a new group of icons. Click an icon to perform a paint effects shortcut.



Importing and exporting images

You can import and export PICT and MacPaint images with the PICT editor.

Importing an image

To import an image into the PICT editor, follow these steps:

1. **Choose Import Graphics from the File menu.**

   A dialog box appears.

2. **Click the types of files you want to import.**

3. **Locate the graphic file and click OK.**

   The imported image replaces everything in the active window.

   Once an image has been imported, you can edit it.

Exporting an image

You can export any image from the PICT editor as a PICT file. To export an image, follow these steps:

1. **Choose Export Graphics from the File menu.**

   A dialog box appears.

2. **Type a name for the new PICT file.**

3. **Move to the location where you want to store the file and click Save.**

Saving a PICT image

The images you create or edit in the PICT editor are automatically saved as resources in the current stack when you close the editor.

- **To save the changes you've made without closing the PICT editor, choose Save from the File menu.**
- **To revert to the most recently stored version of the image, choose Revert from the File menu.**

Special features of the PICT editor

The PICT editor has several special features that give you more control over editing your images.

Editing colors

If your monitor is set to thousands of colors or higher in the Monitors control panel, you can use the color picker to change colors in the PICT editor's palette.

To make changes to an existing color, follow these steps:

1. **Choose Edit Colors from the Options menu.**

   A dialog box with a color palette appears.

2 **Click the color that you want to change.**

The color is surrounded by a thin black line to indicate that it's selected.

3 **Click the Edit button.**

Or double-click the color.

The color picker dialog box appears.

4 **Select a new color by clicking it on the color wheel.**

Or modify the existing color by adjusting its hue, saturation, brightness, or color composition. (For details on using the color picker, see "Coloring by the Numbers" in Chapter 4.)

5 **Click OK when you're finished.**

Zooming

You use zooming to magnify a section of an image for close-in work. You can magnify an image 2, 4, or 8 times.

You can press a number key (1, 2, 3, or 4) to set the magnification to 1, 2, 4, or 8 times the actual size.

- **To zoom in one level of magnification, choose Zoom In from the Options menu.**

  Or press ⌘-J, or ⌘-click the image with the Pencil tool.

- **To zoom out one level of magnification, choose Zoom Out from the Options menu.**

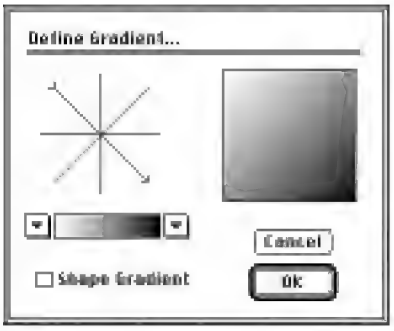
  Or press ⌘-L, or ⌘-Shift-click the image with the Pencil tool.

Creating a gradient

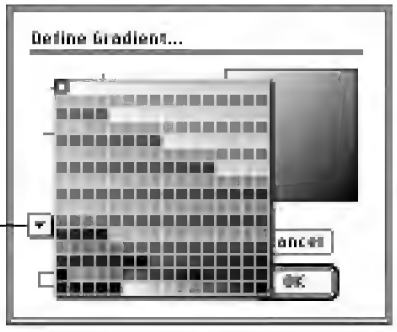
To create a gradient, a combination of two colors that gradually blend as they fill a shape, follow these steps:

1 **Choose Edit Gradient from the Options menu.**

A dialog box appears.



2 **Press the arrow on the left of the gradient bar and choose a color on the pop-up palette.**



Press here to show the color palette.

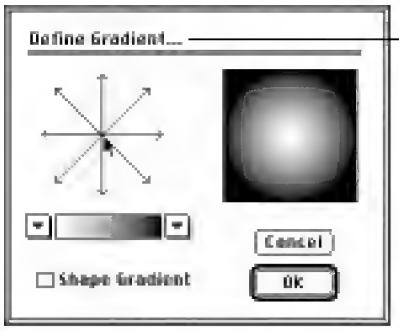
The color you choose becomes the color for the gradient's left side.

3 **Press the arrow to the right of the gradient bar and choose a second color on the pop-up palette.**

The color you choose becomes the color for the gradient's right side.

4 **Click one of the eight lines above the gradient bar, or click the center of the lines once or twice.**

A directional arrow appears to indicate the angle for the gradient. If you click the center, the gradient will radiate in or out from the center (a radial gradient).

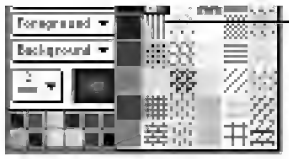


Defining a radial gradient

Click the Shape Gradient box if you want the gradient to adjust to the shape of the object it fills.

5 **When you've finished editing the gradient, click OK.**

- **To use a gradient to fill shapes, select the second pattern on the top row in the Patterns pop-up menu.**


Gradient pattern

The edited gradient appears in the current color box.

Closing the PICT editor

- **To close the PICT editor, choose Quit from the File menu.**

You can also click the Close box on the color paint tools palette.

You return to the standard HyperCard environment.

Maximizing Color Stacks

This chapter describes how to use color stacks efficiently, including getting the most out of memory, conserving disk space, speeding up color operations, and adding color to existing stacks (including stacks created in HyperCard 2.1).

Many of the suggestions listed here involve writing and modifying color scripts. Chapter 9 describes the color scripting commands. For information about scripting in general, see the HyperTalk Reference stack and the *HyperCard Script Language Guide*.

Memory

HyperCard stores in memory the image of each currently visible card (one for each open stack). If the card is black-and-white, HyperCard uses a single bit for each pixel. If your color stack is set for 8-bit color (or 256 colors, the default value), HyperCard uses 24 bits of memory for each color pixel on that card. If you're using 16-bit or 24-bit color, you're using proportionately more memory to display a card.

To use less memory, follow these suggestions:

- Display color at a lower bit depth.
- Reduce the size of your cards.
- Use smaller pictures.
- Open only one color stack at a time.

Disk space

A stack with colorized rectangles, buttons, and fields takes up only slightly more disk space than a non-colorized one. The internal description required for a colorized button, for example, is only 12 bytes. A colorized rectangle requires 18 bytes.

PICT images, on the other hand, take up a lot of space either in a disk file or in a resource added to a stack.

To save disk space, follow these suggestions:

- Use compressed QuickTime pictures instead of PICT images.
- Use Draw-style pictures, which are smaller than bitmaps.

Speed

Adding colorized objects and pictures can slow a stack because of the time it takes to redraw the screen (especially when something is moved under script control).

Bit depth

The greater the bit depth of your color, the slower the stack. It takes up to 24 times longer to draw in 8-bit color than to draw in black-and-white. Higher bit depths take correspondingly longer times.

Changing a card

Each time you change a colorized card by showing or hiding it or by moving colorized elements, the card must be recolored with a call to `addColor colorCard` or some other appropriate command. Recoloring the card is time-consuming.

If you must add color to the card, consider using temporary colors (that is, use `colorField` instead of `addField`).

Changes such as adding text to a field also slow the stack because they require calls to the Color Tools XCMD. The fewer such calls you make, the faster your stack operates.

Whenever possible, use colorCard instead of `colorCardLayered`. (`colorCardLayered` takes twice as long to execute.)

If your color information doesn't change, embed the values in your scripts.

Moving from card to card

When you move to a new colorized card, all the color objects need to be drawn, slowing stack operation.

To speed the stack, use fewer and simpler objects. If objects on a card don't change, use a single card-sized picture, which is usually faster than drawing lots of objects separately.

Whenever possible, remove the `addColor` calls in `openCard` handlers. For example if you have no card layer color and no changing colorized buttons or fields, you need to draw color only when you open the background. (However, you won't get a transition effect when you change cards.)

Types of object

Color rectangles are drawn faster than buttons and fields. PICT images take longest to draw because the data must be read from a resource or a file before being drawn.

PICT size

Large PICT images take longer to draw than small ones. You can make PICT images smaller by removing excess border areas in the PICT editor.

Number of objects

Whenever possible, combine PICT images. It takes longer to draw two color overlays than it does to draw one overlay covering the same area.

Coloring existing stacks

When you open the color editor for the first time in any stack, a number of calls are added to that stack's script. To color existing stacks by hand, use the following calls in their respective system message handlers:

call	handler
install	openStack
colorCard	openCard
lock screen	closeCard
remove	closeStack
pass *messageName*	*all handlers*

Troubleshooting

If the card image (text and buttons) appears before color is updated, you may be missing a `lock screen` message.

If you don't see color, you probably have an extra `lock screen` call.

When you move between cards or to another stack with the screen locked and you see color appear or disappear, you may have to add a call to `lock messages`.

If you see black-and-white images jerkily replaced by colorized images, make sure you've replaced HyperCard's visual effect commands with color transition effects.

Use the message watcher to see what's getting called and what's not getting called.

Scripting for Color

Anything you can do using the commands in the color editor you can also do in your scripts. This chapter describes the scripting functions that the Color Tools add to HyperTalk, HyperCard's scripting language.

All commands except `sort` work with HyperCard 2.1 and higher.

See Chapter 3 of the *HyperCard Reference Manual* for a tutorial introduction to HyperTalk. See the *HyperCard Script Language Guide* for details about scripting and about all the rest of the commands, properties, and functions in HyperTalk.

Syntax terms and conventions used in this chapter

The syntax descriptions in this chapter use the following conventions:

- Words and punctuation in non-italic courier type `like this` should be typed exactly as they appear.
- Words in italic type *like this* are parameter placeholders. You need to replace these words with specific instances. The description for each function makes clear what instances are appropriate.
- Curly braces { } enclose a pair of placeholders from which you must choose. The choices are separated by a vertical bar |.
- Square brackets [ ] enclose optional parameters.
- *Bevel* is a number between 0 and 6, inclusive, representing the width in pixels of a bevel on a button, field, or rectangle.
- *Color* is a set of three RGB numbers that together describe a single color. See Chapter 4 for information on RGB numbers.
- *Index* is the color layer that an overlay occupies (if getting the value), or that you want an overlay to follow (if setting the value). Newly created overlays are initially drawn at the very front color layer. Passing a value of 0 (zero) creates the color overlay all the way in the back; passing a value of -1 creates the overlay at the very front. Each color layer holds a single overlay.
- *Pict* is the name of a PICT image, including those created with the color paint tools.
- *Pt* is a pair of numbers, separated by a comma, describing a point relative to the upper-left corner of the card.
- *Rect* is a set of four numbers, separated by commas, describing a rectangle relative to the upper-left corner of the card. The numbers represent the distance in pixels between
  - left edge of the card and the rectangle's left edge
  - top of the card and the rectangle's top edge
  - left edge of the card and the rectangle's right edge
  - top of the card and the rectangle's bottom edge

- *Effect* is any one of the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

- Finally, a color overlay is an item created or placed through a command in the color editor or by any addColor command. Such items include
  - color on a button or field
  - a color rectangle
  - any PICT image displayed on the screen (except those displayed by HyperTalk's Picture XCMD), including those created with the color paint tools

The addColor XCMD

AddColor is the main external command (XCMD) that makes the Color Tools work. It's installed into every stack that you colorize. You use the addColor XCMD to create scripts that colorize a stack while it's running.

AddColor anatomy

Most of the color commands are actually parameters for addColor. The general structure of an addColor command is as follows:

`addColor` *function, layer, parameter list*

where *function* is any addColor function, *layer* is either cd (for card) or bg (for background), and *parameter list* is one or more parameters associated with a specific function.

The parameters must be entered in exactly the order shown. There are no default positional values. If you don't set all non-optional parameters, the script fails with the error returned in the HyperCard function `the result`.

If a single parameter has several items associated with it and you're using a literal for the items list (as opposed to a single variable), the items list must be enclosed within quotation marks with items separated by commas. For example, in the syntax line

`addColor changeObjectColor,{cd|bg},` *index, color*

color takes three RGB numbers—one each for red, green, and blue:

```
addColor changeObjectColor, cd, 3, "65535, 32767, 16384"
```

The code can also read as follows:

```
put "65535, 32767, 16384" into theColor
addColor changeObjectColor, cd, 3, theColor
```

Check the result: AddColor returns values as well as errors in the HyperCard function `the result`. Because all color tools errors are silent, `the result` is a particularly valuable debugging aid.

AC_RemoteInstall

```
AC_RemoteInstall [logicalExpr]
```

```
AC_RemoteInstall
```

```
AC_RemoteInstall true
```

logicalExpr resolves to either TRUE or FALSE.

This command installs the addColor XCMD under script control. Using it has the same effect as choosing Open Coloring Tools from the Color menu in a stack for the first time.

You use the optional parameter *logicalExpr* to determine if the scripts of the current stack should be modified. The preset value is TRUE. If you set this parameter to FALSE, the addColor resources are installed in the new stack but the stack's scripts remain unaffected.

The following handler (taken from the Color Tools' stack script) installs the addColor XCMD into a new stack and colors a button in the stack red.

```
on SetUpNewStack
  AC_RemoteInstall
  AddColor "addButton","cd",1,"65535,0,0",6,-1
end SetUpNewStack
```

addButton

```
addColor addButton,{cd|bg},ID,color,bevel,index
```

```
addColor addButton,cd,1,"20000,40000,30000",3,-1
```

```
addColor addButton,bg,ID of bg btn "Choices", ¬
    theColor,2,3
```

This command adds a color overlay of the RGB color *color* with beveling level *bevel* to the button whose ID is *ID* on the current card or background. The overlay is initially placed in the frontmost color layer, and moves to color layer *Index* at the next call to `colorCard`.

The following handlers create an orange oval button with a bevel of 3.

```
on newColorButton
  makeNewButton
  addColor addButton, cd, ID of last button,¬
   "65535,32767,16384",3,1
  addColor colorCard -- puts overlay at proper level
end newColorButton

on makeNewButton
  lock screen
  set userLevel to 4 -- to make Objects menu available
  doMenu "New Button"
  set showName of last card button to false
  set the style of the last card button to oval
  set height of last card button to 100
  set width of last card button to 100
  choose Browse tool -- to deselect the new button
  unlock screen
end makeNewButton
```

Also see: `colorButton`
`getButtonIndex`
`removeButton`

addField

```
addColor addField,{cd|bg},ID,color,bevel,index
```

```
addColor addField,cd,1,"20000,40000,30000",3,-1
```

```
addColor addField, bg, ID of fld "Names",¬
    theColor, 2, 3
```

This command adds a color overlay of the RGB color *color* with beveling level *bevel* to the card or background field whose ID is *ID*. The overlay is initially placed in the frontmost color layer, and moves to color layer *index* at the next call to colorCard.

The following handlers create a yellow opaque background field with a bevel of 3. (The wideMargins property is set to true to allow extra margin room for the bevel.)

```
on newColorField
  makeNewField
  put "65535,65535,39321" into yellow
  addColor addField, bg, ID of last field, yellow,3,1
  addColor colorCard -- puts overlay at proper level
end newColorField

on makeNewField
  lock screen
  doMenu "Background"
  set userLevel to 4 -- to make Objects menu available
  doMenu "New Field"
  set wideMargins of last field to true
  doMenu "Background"
  choose Browse tool -- to deselect the new field
  unlock screen
end makeNewField
```

Also see: colorField
getFieldIndex
removeField

addPict

```
addColor addPict,{cd|bg},pict,{pt|rect},{t|o},index
```

```
addColor addPict,cd,"new one","0,0",t,3

addColor addPict,bg,"test pict","20,20,100,100",o,-1
```

This command locates the PICT resource named *PICT* in the current stack and displays it on the card or background. When you specify a point (*pt*), the PICT image appears full-sized with its top-left corner at that point; when you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

The PICT image is initially placed in the frontmost color layer, and moves to color layer *Index* at the next call to colorCard.

When you set the opacity parameter to `t`, the white portions of the PICT image are transparent and anything below the PICT image shows through; when you set it to `o`, the entire PICT image is opaque.

The following handler asks the user for the name of a PICT image, and then places the PICT image on the screen full-sized with its upper-left corner at the pointer position.

```
on pictResource
  ask file "What picture should I use?" of type PICT
  addColor addPict,cd,it,the mouseLoc,o,0 -- in the back
  addColor colorCard -- puts overlay at proper level
end pictResource
```

Also see: `addPictFile`
`colorPict`
`colorPictFile`
`getPictName`
`removeObject`

addPictFile

```
addColor addPictFile,{cd|bg},pictFile,{pt|rect},¬
   {t|o}, index
```

```
addColor addPictFile,cd,"house","0,0",t,3

addColor addPictFile,bg,team,"20,20,100,100",o,-1
```

This command locates the named PICT file and displays it on the card or background. (To display a PICT resource, use `addPict`.) When you specify a point (*pt*), the PICT image appears full-sized with its top-left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

The PICT image is initially placed in the frontmost color layer, and moves to color layer *Index* at the next call to `colorCard`.

When you set the opacity parameter to `t`, the white portions of the PICT image are transparent and anything below the PICT image shows through; when you set it to `o`, the entire PICT image is opaque.

IMPORTANT Your PICT files (or aliases to them) must be in the same folder as HyperCard, Home, or your stack. Do not rename PICT files that you are displaying; HyperCard looks for PICT files by name.

The following handler asks the user for the name of a PICT image, and then displays the PICT image full-sized with its upper-left corner at the pointer position.

```
on pictureOnDIsk
  ask "What picture should I use?"
  addColor addPictFile,cd,it,the mouseLoc,o,0
  addColor colorCard -- puts overlay at proper level
end pictureOnDisk
```

Also see: `addPict`
`colorPict`
`colorPictFile`
`getPictName`
`removeObject`

addRect

```
addColor addRect,{cd|bg},rect,color,bevel,index
```

```
addColor addRect,cd,"0,0,100,100","30000,40000,50000",3,0

addColor addRect,bg,myRectangle,myColor,2,-1
```

This command creates a color rectangle in the RGB color *color* with beveling level *bevel* in the current card or background. The rectangle appears in a size and position specified by *rect.* It is initially placed in the frontmost color layer, and moves to color layer *Index* at the next call to colorCard.

The rectangle is a pure color object.

The following handler adds a red rectangle to color the current background.

```
on fillBack
  put the rect of this cd into theRect
  put "65535,0,0" into theColor
  put 4 into theBevel
  put 0 into theLayer -- 0 for backmost

  addColor addRect,bg,theRect,theColor,theBevel,theLayer
  if the result is not empty then
    answer "Error adding a Rectangle to Database:"¬
    & return &    the result
    exit fillBack
  end if
end fillBack
```

Also see: colorRect
removeObject

changeObjectBevel

```
addColor changeObjectBevel,{cd|bg},index,bevel
```

```
addColor changeObjectBevel,cd,theIndex,newBevel

addColor changeObjectBevel,bg,2,5
```

This command changes the beveling level of the color overlay at color layer *index* on the current card or background to beveling level *bevel*.

If *bevel* is 0, any beveling already assigned to the overlay is removed.

The change to the overlay won't be visible until you make a call to colorCard.

IMPORTANT This command will not create an overlay where none exists, nor will it add a bevel to a transparent button or field.

The bevel is a quality of the color overlay and not of a button or field with which the overlay is associated.

The following handler changes the bevel of the card object under the pointer.

```
on changeBevel
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  put item 1 of the result into theIndex
  if theIndex is -1 then
    answer "There's no bevel to change."
    exit changeBevel
  end if
  addColor getObjectBevel,cd,theIndex
  put the result into oldBevel
  repeat
    put random(7) - 1 into newBevel
    if oldBevel <> newBevel then exit repeat
  end repeat
  addColor changeObjectBevel,cd,theIndex,newBevel
  addColor colorCard
end changeBevel
```

Also see: `getObjectBevel`

changeObjectBounds

```
addColor changeObjectBounds,{cd|bg},index,rectangle
```

```
addColor changeObjectBounds,cd,4,"10,10,100,100"
```

```
addColor changeObjectBounds,bg,theIndex,newRectangle
```

This command changes the rectangle of the PICT image or color rectangle on the current card or background in the specified color layer (*index*). It has no effect on buttons or fields or on their associated color overlays.

Changes to the rectangle won't be visible until you make a call to colorCard.

The following handler moves the card object under the pointer 1/2 inch down and to the right.

```
on changeBounds
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  get the result

  put item 1 of it into theIndex
  put item 2 of it into theType
  if theType < 3 then --Make sure object is right type
    answer "There's no rectangle or PICT image here."
    exit changeBounds
  end if

  addColor getObjectBounds,cd, theIndex
  get the result
  repeat with i = 1 to 4
    add 36 to item i of it -- 36 pixels = 1/2 inch
  end repeat

  addColor changeObjectBounds,cd,theIndex,it--Set new rect
  addColor colorCard
end changeBounds
```

Also see: `getObjectBounds`

changeObjectColor

```
addColor changeObjectColor,{cd|bg},index,color
```

```
addColor changeObjectColor,cd,4,"30000,40000,60000"
```

```
addColor changeObjectColor,bg,theIndex,newColor
```

This command changes the color of the overlay in color layer *index* on the current card or background.

IMPORTANT This command, which applies only to buttons, fields, and rectangles, will not create an overlay where none exists.

The following handler assigns a random color to the object under the pointer.

```
on changeColor
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  put item 1 of the result into theIndex

  repeat with theItem = 1 to 3
    put random(65535) into item theItem of theColor
  end repeat

  addColor changeObjectColor,cd,theIndex,theColor
  addColor colorCard
end changeColor
```

Also see: `getObjectColor`

changeObjectTransparency

```
addColor changeObjectTransparency,{cd|bg},index,{t|o}
```

```
addColor changeObjectTransparency,cd,4,t
```

```
addColor changeObjectTransparency,bg,theIndex,o
```

This command sets the opacity of the PICT image in the designated color layer on the current card or background to either opaque (o) or transparent (t).

When a PICT image has its opacity parameter set to t, the white portions of the PICT image are transparent and any color below the PICT image shows through. When the opacity parameter is set to o, the entire PICT image is opaque.

The following handler toggles the transparency of the PICT image under the pointer.

```
on changeTransparency
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  get the result
  if item 2 of it < 4 then
    answer "The pointer isn't over a PICT image."
    exit changeTransparency
  end if

  put item 1 of it into theIndex
  addColor getObjectColor, cd, theIndex
  get the result
  if item 4 of it is "t"
  then put "o" into opacity
  else put "t" into opacity

  addColor changeObjectTransparency,cd,theIndex,opacity
  addColor colorCard
end changeTransparency
```

colorBackground

```
addColor colorBackground [,effect [,duration]]
```

```
addColor colorBackground
```

```
addColor colorBackground,dissolve,60
```

This command turns off color in the card level and colorizes the background with an optional transitional effect.

When you use a transition effect, you can also specify the duration of the transition in 60ths of a second. The default duration is one second. Transition effects include the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

Use the command `addColor colorCard` to redraw the card colors.

The following handler, which assumes you have colorized items in both the card and background, turns the card color off and on several times.

```
on flashCard
  repeat 3
    addColor colorBackground, dissolve
    addColor colorCard, dissolve
  end repeat
end flashCard
```

Also see: `colorCard`
`colorCardLayered`

colorButton

```
addColor colorButton,{cd|bg},ID,color,bevel
```

```
addColor colorButton,cd,1,"20000,40000,30000",3

addColor colorButton,bg,ID of bg btn "Choices",theColor,2
```

This command adds a temporary color overlay in the RGB color *color* with beveling level *bevel* to the card or background button whose ID is *ID*. The overlay is placed in the frontmost color layer, and is removed at the next call to colorCard or colorBackground.

IMPORTANT Even though the color overlay is drawn at the very front, you still need to specify whether the button is on the card or on the background.

The color you add to a button with this command is temporary. To add color to a button permanently, use the command `addColor addButton`.

The following handler colorizes three buttons with IDs 1, 2, and 3 on the card layer, maintains the color for two seconds, and then removes the color.

```
on tempButton
  repeat with buttonID = 1 to 3
    put "65535, 32767, 16384" into theColor
    put random(7)-1 into theBevel
    addColor colorButton, cd, buttonID, theColor, theBevel
  end repeat
  wait 120
  addColor colorCard
end tempButton
```

Also see: `addButton`
`getButtonIndex`
`removeButton`

colorCard

```
addColor colorCard [,effect [,duration]]
```

```
addColor colorCard
```

```
addColor colorCard,dissolve,60
```

This command colorizes the current card, displaying all color overlays in their proper color layers.

You can also use this command to remove any PICT images or other color overlays temporarily added to the current card or background using `colorButton`, `colorField`, `colorRect`, `colorPict`, or `colorPictFile`.

You can specify a transition effect as the coloring appears on the card, and a duration for the transition in 60ths of a second (ticks). Transition effects include the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

Because it is needed to redraw the screen properly displaying colors where they belong, this command appears in nearly every color handler.

ColorCard draws items on the screen in the following order:

1. all background objects, PICT images, and color rectangles
2. all card objects, PICT images, and color rectangles
3. HyperCard's black-and-white image

The following handler temporarily draws a blue rectangle on the screen, waits for the mouse button to be pressed, and then destroys the rectangle with a call to `colorCard`.

```
on waitForIt
  addColor colorRect, cd, "30,30,200,200", "0,0,65000", 6
  wait until the mouse is down
  addColor colorCard
end waitForIt
```

IMPORTANT To avoid the sudden startling appearance of color objects on the screen, lock the screen before you change cards; then go to the destination card and call `addColor colorCard`.

Also see: `colorBackground`
`colorCardLayered`

colorCardLayered

```
addColor colorCardLayered [,effect [,duration]]
```

```
addColor colorCardLayered
```

```
addColor colorCardLayered, dissolve,60
```

This command colorizes the current card, displaying all color overlays in their proper color layers.

You can also use this command to remove any overlays temporarily added to the current card or background using `colorButton, colorField, colorRect, colorPict,` or `colorPictFile.`

You can specify a transition effect as the coloring appears on the card, and a duration for the transition in 60ths of a second (ticks). Transition effects include the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

ColorCardLayered colors items in the following order, which is different from the order that `colorCard` uses:

1. Background PICT images and color rectangles
2. Card PICT images and colorized rectangles
3. Background button overlays and field overlays
4. Card button overlays and field overlays
5. HyperCard's black-and-white image

colorCardLayered is slow: It takes twice as long to colorize a card using this command as it does using `colorCard`.

The following handler shows how card layering works.

```
on showColoringSequence
  put "0,65000,0" into green
  put "65000,0,0" into red
  put "65000,65000,30000" into yellow
  put "0,0,65000" into blue

  put ID of last cd btn into lastCdBtnID
  put ID of last bg btn into lastBgBtnID

  addColor addButton, cd, lastCdBtnID, green, 3, -1
  addColor addButton, bg, lastBgBtnID, yellow, 3, -1
  addColor addRect, cd, "0,0,100,100", red, 5, -1
  addColor addRect, bg, "100,100,200,200", blue, 5, -1
  addColor remove
  addColor install
  addColor colorCardLayered, dissolve, 60

end showColoringSequence
```

Also see: `colorCard`
`colorBackground`

colorField

```
addColor colorField,{cd|bg},ID,color,bevel
```

```
addColor colorField,cd,1,"20000,40000,30000",3

addColor colorField,bg,ID of fld "Animals",theColor,2
```

This command adds a temporary color overlay in the RGB color *color* with beveling level *bevel* to the field whose ID is *ID* in the specified card or background domain. The overlay is placed in the frontmost color layer and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

IMPORTANT Even though the color overlay is drawn at the very front, you still need to specify whether the field is on the card or on the background.

The color you add to a field with this command is temporary. To add color to a field permanently, use the command `addColor addField`.

The following handler colorizes three background fields with IDs 1, 2, and 3, maintains the color for two seconds, and then removes it.

```
on tempField
  repeat with fieldID = 1 to 3
    put "65535, 0, 0" into theColor
    put random(7)-1 into theBevel
    addColor colorField, cd, fieldID, theColor, theBevel
  end repeat
  wait 120
  addColor colorCard
end tempField
```

Also see: `addField`
`getFieldIndex`
`removeField`

colorPict

```
addColor colorPict,{cd|bg},pict,{pt|rect},{t|o},¬
   [,effect [,duration]]
```

```
addColor colorPict,cd,"new one","0,0",t
```

```
addColor colorPict,bg,"test pict","20,20,100,100",o,¬
    fromTopLeft
```

This command locates the PICT resource named *PICT* in the current stack and temporarily displays it on the current card or background. When you specify a point (*pt*), the PICT image appears full-sized with its top left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

You can specify a transition effect as the PICT image appears on the screen, and a duration for the transition in 60ths of a second (ticks). Transition effects include the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

The PICT image appears in the frontmost color layer, and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

IMPORTANT Even though the PICT image is drawn at the very front, you still need to specify whether the PICT image is on the card or on the background.

A PICT image you display with this command is temporary. If you want to display a PICT resource each time you open the card, use the command `addColor addPict`.

When you set the opacity parameter to t, the white portions of the PICT image become transparent and any color below the PICT image shows through. When you set the opacity parameter to o, the entire PICT image is opaque.

The following handler temporarily puts a PICT resource on the screen with its upper-left corner at the pointer. The PICT image is removed when the mouse button is pressed.

```
on tempPictResource
  ask "What picture should I use?"
  addColor colorPict,cd,it,the mouseLoc,o,irisOpen
  wait until the mouse is down
  addColor colorCard
end tempPictResource
```

Also see: `addPict`
`addPictFile`
`colorPictFile`
`getPictName`
`removeObject`

colorPictFile

```
addColor colorPictFile,{cd|bg},pict,{pt|rect},{t|o},¬
   [,effect [,duration]]
```

```
addColor colorPictFile,cd,"new one","0,0",t
```

```
addColor colorPictFile,bg,"test pict","20,20,100,100",o,¬
    fromTopLeft
```

This command temporarily displays a copy of the PICT file named *pict*. (To temporarily display a PICT resource, use `colorPict`.) When you specify a point (*pt*), the PICT image appears full-sized with its upper-left corner at that point. When you specify a rectangle (*rect*), the PICT image is scaled to fit within the rectangle.

You can specify a transition effect as the PICT image appears on the screen, and a duration for the transition in 60ths of a second (ticks). Transition effects include the following:

fromLeft	fromRight	fromTop
fromBottom	fromTopLeft	fromTopRight
fromBottomLeft	fromBottomRight	dissolve
irisOpen	irisClose	checkerBoardOpen
checkerBoardClose	circleCheckerOpen	circleCheckerClose
barnDoorOpen	barnDoorClose	combVertical
combHorizontal	rectOpen	rectClose
venetianBlindsHorizontal	venetianBlindsVertical	rakeHorizOpen
rakeHorizClose	rakeVertOpen	rakeVertClose

The PICT image appears in the frontmost color layer, and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

IMPORTANT Even though the PICT image is drawn at the very front, you still need to specify whether the PICT image is on the card or on the background.

A PICT image you display with this command is temporary. If you want to display a PICT file each time you open the card, use the command `addColor addPictFile`.

When you set the opacity parameter to t, the white portions of the PICT image become transparent and any color below the PICT image shows through. When you set the opacity parameter to o, the entire PICT image is opaque.

The following handler temporarily displays a copy of a PICT file at the upper-left corner of the card. When the screen is redrawn, the PICT image will disappear.

```
on tempPictFile
  ask "What PICT file should I use?"
  put it into fileName
  addColor colorPictFile, cd, fileName, "0,0",t
end tempPictfile
```

Also see: `addPict`
`addPictFile`
`colorPict`
`getPictName`
`removeObject`

colorRect

```
addColor colorRect,{cd|bg},rect,color,bevel
```

```
addColor colorRect,cd,"0,0,100,100","30000,40000,50000",3

addColor colorRect,bg,myRectangle,myColor,2
```

This command temporarily creates a color rectangle on the card or background in the RGB color *color* with beveling level *bevel*. The rectangle appears in a size and position specified by *rect*.

The rectangle is displayed in the frontmost color layer, and is removed at the next call to `colorCard`, `colorBackground`, or `colorCardLayered`.

IMPORTANT Even though the color overlay is drawn at the very front, you still need to specify whether the rectangle is on the card or on the background.

The rectangle that you display with this command is temporary. To display a color rectangle each time you open the card, use the command `addColor addRect`.

The rectangle is a pure color object.

The following handler paints color rectangles in random locations on the card until the mouse button is pressed; then the rectangles are all erased.

```
on randomTempRects
  repeat until the mouse is down
    repeat with color=1 to 3
      put random(65535) into item color of theColor
    end repeat

    put random(7)-1 into theBevel
```

handler continues ▶

```
    put random(100) into item 1 of theRect
    put random(100) into item 2 of theRect
    put item 3 of the rect of this card into btm
    put random(btm) + item 1 of theRect¬
    into item 3 of theRect
    put item 4 of the rect of this card into rgt
    put random(rgt) + item 2 of theRect¬
    into item 4 of theRect

    AddColor colorRect,cd,theRect,theColor,theBevel
  end repeat
  --            Erase 'em all
  addColor colorCard
end randomTempRects
```

Also see: addRect
removeObject

compact

```
addColor compact,{cd|bg}
```

```
addColor compact,bg

addColor compact,cd
```

This command performs internal housecleaning on the Color Tools' internal database, removing information about deleted colorized buttons and fields.

To keep color information as up-to-date as possible, use this command whenever you delete a button or field you've colorized.

You must issue this command separately for the card and the background, as demonstrated in the following handler:

```
on killButtons
    delete card button "Green"
    delete background button "Blue"
    addColor compact, bg
    addColor compact, cd
    addColor colorCard
end killButtons
```

The `compact` command doesn't remove the color of the deleted object from the screen. To do that, you must make a call to `colorCard` or `colorBackground`, as appropriate.

The command `addColor compact` is not the same as `Compact Stack`. Using `addColor compact` compacts the color database but does not compact the stack. To compact the stack, use the standard HyperTalk command `doMenu Compact Stack` with the Browse, Field, or Rectangle tool selected.

The following handler deletes a field from the background and properly cleans up after itself.

```
on killFieldCleanly
  delete background field "Redundant"
  addColor compact, bg
  addColor colorCard
  if the freeSize of this stack¬
   >.05 * the size of this stack then
    get the tool -- what tool is in use?
    choose the browse tool
    doMenu "Compact Stack"
    choose it -- restore user's tool
  end if
end killFieldCleanly
```

disable

```
addColor disable
```

```
addColor disable
```

This command temporarily turns color tools off, preserving the off-screen memory buffers that hold color information. You use this command to prevent the color tools from interfering with an XCMD that you're about to run.

IMPORTANT To turn color tools back on again, use the command `addColor enable`. Until you do so, all other addColor calls are ignored.

The following example turns off the color tools, runs an XCMD, and then turns the color tools back on:

```
on doAnotherXCMD
    addColor disable
    runMyZippyXCMD
    addColor enable
end doAnotherXCMD
```

The following handler attempts to color a button orange with the color tools turned off, then turns the tools on again to complete the job:

```
on stopTheShow
    addColor disable
    addColor addButton,cd,ID of btn "Books",¬
    "65535,32767,16384",3,0
    addColor colorCard
    answer "Nothing works yet.."

    addColor enable
    addColor addButton,cd,ID of btn "Books",¬
    "65535,32767,16384",3,0
    addColor colorCard
    answer "...but now it does."
end stopTheShow
```

Also see: `enable`
`disableObject`
`remove`

disableObject

```
addColor disableObject,{cd|bg},index
```

```
addColor disableObject,cd,3
```

```
addColor disableObject,bg,theColorLayer
```

This command temporarily turns off an overlay's ability to show itself.

Only the color overlay is hidden. The button or field is still visible. To hide a button or field, use HyperTalk's `hide` command.

IMPORTANT Before you disable an overlay, note its index value. While the overlay is disabled you can't address it to learn its index, a value you'll need later to restore the overlay.

The following example hides the color overlay on the card under the pointer while you hold down the mouse button.

```
on hideWhenDown

  addColor getObjectClicked,cd,the mouseLoc
  put item 1 of the result into theIndex

  wait until the mouse is down
  addColor disableObject,cd,theIndex
  addColor colorCard

  wait until the mouse is up
  addColor enableObject,cd,theIndex
  addColor colorCard

end hideWhenDown
```

Also see: `enableObject`
`disable`
`remove`

enable

```
addColor enable
```

```
addColor enable
```

This command turns the color tools on after they have been made inoperative by `disable`.

The following example turns off the color tools, runs an XCMD, and then turns the color tools back on:

```
on doAnotherXCMD
    addColor disable
    runMyZippyXCMD
    addColor enable
end doAnotherXCMD
```

Choosing Open Coloring Tools from the Color menu also turns the coloring tools back on.

The following handler attempts to color a button red with the color tools turned off, then turns the tools on again to complete the job:

```
on stopTheShow
    addColor disable
    addColor addButton, cd, ID of btn "CDs",¬
    "65535,0,0",3,0
    addColor colorCard
    answer "Nothing works yet.."

    addColor enable
    addColor addButton,cd, ID of btn "CDs",¬
    "65535,0,0",3,0
    addColor colorCard
    answer "but now it does."
end stopTheShow
```

Also see: `disable`
`enableObject`
`install`

enableObject

```
addColor enableObject,{cd|bg},index
```

```
addColor enableObject,cd,3
```

```
addColor enableObject,bg,theColorLayer
```

This command turns on a color object previously turned off by `disableObject`.

To display the object after you turn it on, use `addColor colorCard`.

The following handler hides the color of the card object under the pointer while the mouse button is down.

```
on showWhenUp
    addColor getObjectClicked,cd,the mouseLoc
    put item 1 of the result into theIndex

    wait until the mouse is down
    addColor disableObject,cd,theIndex
    addColor colorCard

    wait until the mouse is up
    addColor enableObject,cd,theIndex
    addColor colorCard
end showWhenUp
```

Also see: `enableObject`
`disable`
`install`

getBitsCall

```
addColor getBitsCall
```

```
addColor getBitsCall
```

This command, designed for XCMD developers, returns the address of the addColor patch in HyperTalk's function `the result`. The address provides access to the Color Tools' color buffers for the current window.

The following handler displays the patch address in an answer dialog box.

```
on findPatchAddress
    addColor getBitsCall
    answer "Color buffers start at" && the result & "."
end findPatchAddress
```

getButtonIndex

```
addColor getButtonIndex,{cd|bg},ID
```

```
addColor getButtonIndex,cd,3
```

```
addColor getButtonIndex,bg,ID of button "Belly"
```

This command returns the index value of the color overlay for the card or background button whose ID is *ID*.

The higher the number of the index value, the closer the color overlay is to the front of the color layer.

A value of –1 means that the specified button has no color overlay.

The following handler reports the index value for the button under the pointer.

```
on locateButtonColor
  get the mouseLoc
  -- Is it a card button?
  repeat with here = number of card buttons down to 1
    if it is within the rect of card button here then
      get the Id of card button here

      addColor getButtonIndex,cd,it

      answer "The card button at the pointer is at index"¬
      && the result && "."
      exit locateButtonColor
    end if
end repeat

--Is it a background button?
  repeat with here= number of background buttons down to 1
    if it is within the rect of bg button here then
      get the Id of background button here
      addColor getButtonIndex,bg,it
```

handler continues ▶

```
answer "The button at the pointer is at index"¬
      && the result && "."
      exit locateButtonColor
    end if
  end repeat
end locateButtonColor
```

Also see: addButton
colorButton
removeButton

getFieldIndex

```
addColor getFieldIndex,{cd|bg},ID
```

```
addColor getFieldIndex,cd,3
```

```
addColor getFieldIndex,bg,ID of field "Elysium"
```

This command returns the index value of the color overlay for the card or background field whose ID is *ID*.

The higher the number of the index value, the closer the color overlay is to the front of the color layer.

A value of –1 means that the specified field has no color overlay.

The following handler reports the index value for the field under the pointer.

```
on locateFieldColor
  get the mouseLoc
  -- Is it a card field?
  repeat with here = number of card fields down to 1
    if it is within the rect of card field here then
      get the Id of card field here

addColor getFieldIndex,cd,it

answer "The card field at the pointer is at index"¬
      && the result && "."
      exit to HyperCard
    end if
  end repeat

-- Is it a background field?
  repeat with here= number of background fields down to 1
    if it is within the rect of bg field here then
      get the Id of background field here
      addColor getFieldIndex,bg,it
```

handler continues ▶

```
answer "The field at the pointer is at index"¬
      && the result && "."
      exit to HyperCard
    end if
  end repeat
end locateFieldColor
```

Also see: addField
colorField
removeField

getObjectBevel

```
addColor getObjectBevel,{cd|bg},index
```

```
addColor getObjectBevel,cd,3
```

```
addColor getObjectBevel,bg,theIndex
```

This command returns the bevel width in pixels of the color overlay for the card or background button, field, or rectangle at color layer *index*. The bevel width is returned in the Hypercard function `the result`.

Meaningful widths are 0 through 6. If `the result` is empty, you've provided an invalid index number.

The following handler reports the bevel of the card object under the pointer.

```
on getBevel
    addColor getObjectClicked,cd,the mouseLoc
    get item 1 of the result
    addColor getObjectBevel, cd, it
    answer "The bevel is" && the result && "."
end getBevel
```

Also see: `changeObjectBevel`

getObjectBounds

```
addColor getObjectBounds,{cd|bg},index
```

```
addColor getObjectBounds,cd,3

addColor getObjectBounds,bg,theIndex
```

This command returns the bounding rectangle of the card or background color overlay at color layer *index*. The rectangle is returned as a comma-separated list of four numbers in the Hypercard function `the result`.

If `the result` is empty, you've provided an invalid index number.

The following handler reports the rectangle of the card object under the pointer.

```
on getBounds
   addColor getObjectClicked,cd,the mouseLoc
   get the result
    put item 1 of it into theIndex
   addColor getObjectBounds, cd, theIndex
   answer "The rectangle of the object at the pointer is"¬
   && the result & "."
end getBounds
```

Also see: `changeObjectBounds`

getObjectClicked

```
addColor getObjectClicked,{cd|bg},point[,type]
```

```
addColor getObjectClicked,cd,"100,300",buttonType
```

```
addColor getObjectClicked,bg,the mouseLoc
```

This command returns the index value and the type of card or background color overlay at point *point*. The information is returned in the Hypercard function `the result`.

Optionally, you can specify in *type* the type of object that this command reports on. The type you specify must be one of the following:

- buttonType
- fieldType
- rectType
- pictType
- pictFileType

The first number returned indicates the index (that is, the color layer) in which the overlay exists. The second number, separated from the first by a comma, represents the type of overlay located, as follows:

Returned value	Overlay type
1	button
2	field
3	rectangle
4	PICT resource
5	PICT file

A result of –1, –1 means that no color overlay [of the specified type] exists at the location you specified.

The following handler reports the type, location, and index of the card object under the pointer.

```
on getIndex
  put "button,field,rectangle,PICT resource,PICT file"¬
  into theType
  put the mouseLoc into thePoint
  addColor getObjectClicked,cd,thePoint
  get the result
  if item 1 of it is -1 then
    answer "No card overlay there."
    exit getIndex
  end if
  put item (item 2 of it) of theType into theObject
  put item 1 of it into theIndex
  answer "The index of the" && theObject¬
  && "at" && thePoint  && "is" && theIndex & "."
end getIndex
```

getObjectColor

```
addColor getObjectColor,{cd|bg},index
```

```
addColor getObjectColor,cd,3
```

```
addColor getObjectColor,bg,theIndex
```

This command returns a comma-separated list of either three or four values in the HyperTalk function `the result`. The first three values compose the RGB color of the card or background button, field, or rectangle located at color layer *index*. A fourth value is returned when the object is a PICT image; the value is the PICT image's opacity (t for transparent, o for opaque).

If the item is a PICT image, the color values are meaningless.

If `the result` is empty, you've provided an invalid index number.

The following handler reports the color or opacity of the item under the pointer.

```
on getColor
    addColor getObjectClicked,cd,the mouseloc
    addColor getObjectColor,cd,item 1 of the result
    get the result
    if it is empty
    then answer "The index number is invalid."
    else if the number of items in it = 3
    then answer "The object's RGB color is" && it & "."
    else answer "The PICT image's opacity value is"¬
    && item 4 of it & "."
end getColor
```

Also see: `changeObjectColor`

getObjectType

```
addColor getObjectType,{cd|bg},index
```

```
addColor getObjectType,cd,3
```

```
addColor getObjectType,bg,theIndex
```

This command returns the type of overlay located at color layer *index* in the HyperTalk function `the result`. The value returned is one of the following:

- buttonType
- fieldType
- rectType
- pictType
- pictFileType

If `the result` is empty, you've provided an invalid index number.

The following handler reports the type of overlay in the first five layers of color on the current card.

```
on identifyOverlay
    repeat with theIndex = 1 to 5
      addColor getObjectType,cd,theIndex
      put "Layer" && theIndex &":" && the result¬
      into line theIndex of typeList
    end repeat
    answer "Here are the overlay types" &&¬
    "in the first five color layers."¬
    & return & return & typeList
end identifyOverlay
```

getPictName

```
addColor getPictName,{cd|bg},index
```

```
addColor getPictName,cd,3
```

```
addColor getPictName,bg,theIndex
```

This command returns the name of the PICT resource or file located at color layer *index* in the HyperTalk function `the result`.

If `the result` is empty, you've provided an invalid index number.

The following handler reports the name of the PICT resource or file under the pointer.

```
on whatPICT
    addColor getObjectClicked,cd,the mouseloc
    addColor getPictName, cd, item 1 of the result
    get the result
    if it is empty then answer "Sorry - no PICT here."
    else answer "The PICT at the pointer is"¬
    && it &"."
end whatPict
```

install

```
addColor install [, bitDepth]
```

```
addColor install
```

```
addColor install, 24
```

This command makes the color tools usable in the current stack, creating off-screen memory buffers to hold color information. It belongs in the stack script's openStack handler.

The color tools are installed with a preset bit depth of 8, allowing 256 colors; to increase or decrease the bit depth, use the optional *bitDepth* parameter.

This command, which must be called before any other color command, is automatically installed in the stack script when you first open the color editor.

If your monitor is set to a different bit depth, you may not be able to use all of the colors in the palette.

The following handler, placed in the stack script, restricts the current stack to work only in 16 colors (thus reducing the amount of RAM the stack needs).

```
on openStack
  addColor install, 4
end openStack
```

Also see: `remove`
`enable`

moveBackward

```
addColor moveBackward,{cd|bg},index
```

```
addColor moveBackward,cd,3
```

```
addColor moveBackward,bg,theIndex
```

This command moves the card or background color overlay at color position *index* one layer back.

The overlay's new position is returned in the HyperTalk function `the result`, and will be one number lower than the value you specify in *index*. If *index* holds the value 1, indicating that the overlay is already all the way in the back, the returned value will remain 1.

The overlay that holds the immediately lower position before this command is executed moves forward one layer. (In effect, the overlays switch layers.)

Some index values may change: Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler moves the overlay under the pointer one layer closer to the back.

```
on moveBack
  addColor getObjectClicked,cd,the mouseLoc
  put item 1 of the result into theIndex
  if theIndex is -1 then
    answer "There's no color object on the card there."
    exit to HyperCard
  else if theIndex is 1 then
    answer "The object is all the way at the back."
    exit to HyperCard
  end if
  addColor moveBackward,cd,theIndex
  addColor colorCard
end moveBack
```

Also see: `moveForward`
`moveToBack`

moveForward

```
addColor moveForward,{cd|bg},index
```

```
addColor moveForward,cd,3
```

```
addColor moveForward,bg,theIndex
```

This command moves the card or background color overlay at color position *index* one color layer forward.

The overlay's new position is returned in the HyperTalk function `the result`, and will be one number higher than the value in *index*.

The overlay that holds the immediately higher position before this command is executed moves backward one layer. (In effect, the overlays switch layers.)

Some index values may change: Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following button handler moves the overlay under the pointer one layer closer to the front.

```
on moveUp
  addColor getObjectClicked,cd,the mouseLoc
  put item 1 of the result into theIndex
  if theIndex is -1 then
    answer "There's no color object on the card there."
    exit to HyperCard
  end if
  addColor moveForward,cd,theIndex
  addColor colorCard
end moveUp
```

Also see: `moveBackward`
`moveToFront`

moveToBack

```
addColor moveToBack,{cd|bg},index

addColor moveToBack,cd,3

addColor moveToBack,bg,theIndex
```

This command moves the card or background color overlay at color position *index* all the way to the back.

The overlay's new position is returned in the HyperTalk function `the result`. It is always the value 1.

All overlays that hold lower positions before this command is executed move forward one layer.

Some index values may change: Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler rotates five overlapping button color overlays so that each one in turn momentarily occupies the top postion. Hold down the mouse button to exit the handler.

```
on rotateBottomButton
    repeat
      repeat with top = 1 to 5
        addColor getButtonIndex,cd,ID of cd btn top
        addColor moveToBack, cd, the result
        addColor colorCard
        if the mouse is down then exit rotateBottomButton
      end repeat
    end repeat
    beep 3
end rotateBottomButton
```

Also see: `moveBackward`
`moveToFront`

moveToFront

```
addColor moveToFront,{cd|bg},index
```

```
addColor moveToFront,cd,3
```

```
addColor moveToFront,bg,theIndex
```

This command moves the card or background color overlay at color position *index* all the way to the front.

The overlay's new position is returned in the HyperTalk function `the result`.

All overlays that hold higher positions before this command is executed move back one layer.

Some index values may change: Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The result of using this command is most visually apparent when color overlays overlap.

The following handler rotates five overlapping button color overlays so that each one in turn momentarily occupies the top postion. Hold down the mouse button to exit the handler.

```
on rotateTopButton
    repeat
      repeat with top = 1 to 5
        addColor getButtonIndex,cd,ID of cd btn top
        addColor moveToFront, cd, the result
        addColor colorCard
        if the mouse is down then exit rotateTopButton
      end repeat
    end repeat
    beep 3
end rotateTopButton
```

Also see: `moveForward`
`moveToBack`

remove

```
addColor remove
```

```
addColor remove
```

This command frees the memory allocated to color for the current stack. It disposes of all the color information stored in off-screen buffers, and immediately turns off all color in the stack.

To see this command in operation, create a stack with some color on it and enter the command `addColor remove` through the Message box.

IMPORTANT This command is automatically installed in the stack script's closeStack handler when you first open the color editor. It prevents color artifacts from appearing in the next stack you open.

The following handler deallocates the stack's color memory buffer when the stack is closed.

```
on closeStack
    AddColor remove
    pass closeStack
end closeStack
```

Also see: `install`
`disable`

removeButton

```
addColor removeButton,{cd|bg},ID
```

```
addColor removeButton,cd,3
```

```
addColor removeButton,bg,theID
```

This command removes the color overlay from the card or background button whose ID is *ID*. The color disappears from the screen at the next call to colorCard, colorBackground, or colorCardLayered (as appropriate).

The index values of all higher-numbered overlays decrease by one.

IMPORTANT This command removes a button's color overlay. It does *not* remove the button. To remove the button, use HyperTalk's `delete` command.

The following handler removes the color overlay from the card button under the pointer.

```
on dumpButtonColor
  put the mouseLoc into thePoint
  -- What button is it?
  repeat with thisOne = number of card buttons down to 1
    if thePoint is within the rect of cd btn thisOne then
      addColor removeButton, cd, id of cd btn thisOne
      addColor colorCard
      exit dumpButtonColor
    end if
  end repeat
  Answer "That's not a button."
end dumpButtonColor
```

Also see: `addButton`

removeField

```
addColor removeField,{cd|bg},ID
```

```
addColor removeField,cd,3
```

```
addColor removeField,bg,theID
```

This command removes the color overlay from the card or background field whose ID is *ID*. The color disappears from the screen at the next call to colorCard, colorBackground, or colorCardLayered (as appropriate).

The index values of all higher-numbered overlays decrease by one.

IMPORTANT This command removes a field's color overlay. It does *not* remove the field. To remove the field, use HyperTalk's `delete` command.

The following handler removes the color overlay from the card field under the pointer.

```
on dumpFieldColor
  put the mouseLoc into thePoint
  -- What field is it?
  repeat with thisOne = number of card fields down to 1
    if thePoint is within the rect of cd fld thisOne then
      addColor removefield, cd, id of cd fld thisOne
      addColor colorCard
      exit dumpFieldColor
    end if
  end repeat
  Answer "That's not a field."
end dumpFieldColor
```

Also see: `addField`

removeObject

```
addColor removeObject,{cd|bg},index
```

```
addColor removeObject,cd,3
```

```
addColor removeObject,bg,theIndex
```

This command removes the color overlay at color layer *index*. The color disappears from the screen at the next call to `colorCard`, `colorBackground`, or `colorCardLayered` (as appropriate). The index values of all higher-numbered overlays decrease by one.

The following handler removes the color overlay under the pointer whether the overlay is on the card or the background.

```
on dumpColor
  addColor getObjectClicked,cd,the mouseLoc
  get the result
  addColor removeObject,cd,item 1 of it
  addColor colorCard

addColor getObjectClicked,bg,the mouseLoc
  get the result
  addColor removeObject,bg,item 1 of it
  addColor colorCard
end dumpColor
```

Also see: `remove`

sort

```
addColor sort,{cd|bg}
```

```
addColor sort,cd

addColor sort,bg
```

This command, which works only in HyperCard 2.2 or higher, makes the color layering on the card or background match HyperCard's layering. It arranges objects so that buttons and fields are in front of colorized rectangles and PICT images in the card and background layers.

IMPORTANT This command bears no relationship to HyperTalk's `sort` command.

If you've reordered colorized buttons or fields with calls to `moveForward`, `moveBackward`, `moveToFront`, or `moveToBack`, using `addColor sort` properly associates the colors of buttons and fields with their respective objects in the proper layers. The command has no effect, however, on the layering of buttons and fields accomplished through the Bring Closer or Send Farther commands in HyperCard's Objects menu.

Some index values may change: Because index values are based on the color layer that an overlay occupies, the index value of any color layer moved by this command will change.

The following handler sorts the color database for the current card and background.

```
on sortIt
  addColor sort, cd
  addColor sort, bg
end sortIt
```

Appendix A
Troubleshooting

This appendix describes some common areas of confusion and what to do about them.

There's no Color menu in the menu bar.

Look in the Home stack for a button labeled "Color Tools are OFF." If you see it, click it; the Color menu will appear in the menu bar.

If you don't see "Color Tools are OFF," the Color Tools are probably not yet installed. See Chapter 3, "Installing Color Tools," for instructions on preparing HyperCard for color.

My monitor is set to display thousands of colors, but I can't see as many colors as I expected in my stack.

You need to change the stack's openStack handler. Replace the line `addColor install` with `addColor install, 24` or `addColor install` with `addColor install, 24,` depending on the bit depth you want.

Color doesn't show in my stacks or standalone applications, even though I have a color monitor.

Increase the memory allocated to HyperCard and to your standalone applications. Begin with an allocation of 2200K (5120K on HyperCard for Power Macintosh) and repeatedly increase the allocation by 300K until you can see color. For instructions on how to increase memory allocation, see "Preparing a Color Standalone Application" in Chapter 3.

I chose a PICT image to add to my stack using the Import command, but it doesn't appear.

Increase the memory allocated to HyperCard to 4000K or more.

I selected an object in the color editor, but Item Info isn't available.

The object hasn't been colorized yet. Click a color in the palette; then try again.

I can't select an item in the color editor.

You may not have selected the proper icon at the top of the color palette. Click the Button, Field, Pict, Rect, or Paint icon on the color palette; then select an item of the corresponding type.

I turned Color Tools off in the Home stack, but I still see color in my colorized stacks.

Turning Color Tools off doesn't remove color from a stack. When you colorize a stack, the color becomes a permanent part of the stack unless you remove the stack's color resources with an application program such as ResEdit.

I tried to print a colorized stack on my color printer, but none of the colors show.

HyperCard doesn't support printing in color. However, you can purchase software that allows you to print your color stacks.

My HyperCard visual effects have stopped working.

The color editor adds a `closeCard` HyperTalk handler with a `lock screen` command to the stack. This command prevents HyperCard's standard visual effects from working.

While HyperCard's visual effects and the Color Tools' transition effects both work if you remove the `lock screen` command, you probably won't like the results. HyperCard's visual effects work only on HyperCard's black-and-white images, with all color turned off as they operate; if you follow a visual effect with any color effect, the image stutters as color is turned on.

When I move a colorized object, the color doesn't follow it.

Choose Redraw Screen from the Color menu. If you're moving the object under script control, lock the screen before moving it; then recolor the object with a dissolve effect to make the transition smoother.

I colorized a stack and immediately saved it as a standalone application. When I launch the application, it doesn't show any color.

Resources added to a stack are not installed until the stack is closed. After you colorize a stack, you need to close and reopen it before you save it as a standalone application.

Make sure the standalone application has at least 2200K of memory assigned (5120K of memory on HyperCard for Power Macintosh).

I can't shut down the computer.

You must close the color editor or quit HyperCard before using the Shut Down command.

Appendix B
Updates

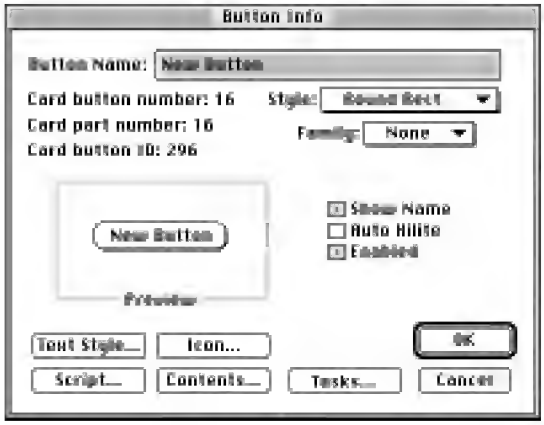
This appendix corrects information that has changed since the rest of the documentation was printed and provides information about features that have been improved since the last version of HyperCard was released.

HyperCard

Button Info dialog box

On computers using Mac OS 7.0 or higher, the Button Info dialog box has been changed to reflect the addition of the Tasks window, described in Chapter 1.

A new button, Tasks, appears in the box, replacing Effects and LinkTo.



HyperTalk

answer folder prompt

The new parameter `folder` displays a dialog box from which you select a folder. The path to the folder, including the trailing colon, is returned in the local variable `it`.

The variable "it" returns "My Disk:HyperCard:Your Tour of HyperCard:"

blindTyping

The *HyperCard Script Language Guide* says that the user level must be set to 5 in order to set the `blindTyping` property. In fact, the user level can be set to any level.

On the Preferences card in the Home stack, however, you can set this property only if the user level is set to 5.

the diskSpace [of disk diskName]

Formerly, you could get the space available only for the current disk. Now you can specify any mounted hard drive or floppy disk.

fade

The card "Tip:List of synonyms" in the Help extras stack lists `fade` as a synonym for the visual effect `dissolve`. This is inaccurate.

A similar error appears on page 219 of the *HyperCard Script Language Guide* in the sample script for `go`.

find international

This form of the `find` command is no longer case-sensitive.

mouseDoubleClick

Checkboxes and radio buttons no longer receive `mouseDoubleClick` messages. Instead they receive `mouseDown`, `mouseStillDown`, and `mouseUp` messages just as they did in versions before HyperCard 2.2.

the stackSpace

The description of the stackSpace function on page 346 of the *HyperCard Script Language Guide* is inaccurate. Instead, use the information in the HyperTalk Reference stack.

the [long] version of scriptingLanguage componentName

The information in the HyperCard AppleScript Reference stack is incomplete.

If you use the option `long`, this property returns an 8-character hexadecimal string that represents the version of the software and the version of its application programming interface (API). For AppleScript 1.1, the hex string is 01100110.

If you do not use the option `long`, this property returns the version of the component as a decimal string. For AppleScript 1.1, the decimal string is 1.10.

You can compare the returned value to an arbitrary number. For example:

```
if the version of scriptingLanguage "AppleScript" >= 1.1
then...
```

quit system message

Page 137 of the *HyperCard Script Language Guide* says that `quit` is sent to the current card immediately following the `closeStack` message when you choose Quit HyperCard from the File menu.

This is true only when there is a single stack open. When more than one stack is open, `quit` is sent to the current card in the last open stack immediately following that stack's `closeStack` message.

HyperTalk Beginner's Guide

The *HyperTalk Beginner's Guide,* referred to on page xix of the *HyperCard Script Language Guide,* is no longer available.

Apple events

picture

`picture` is a new OSA (Open Scripting Architecture) property of card and background objects that returns the bitmap of a card or background as a QuickDraw picture. It lets you pass around picture data so that you no longer need to go into the paint tools to select and copy a picture.

The picture property is read-only.

The following AppleScript example puts the background picture into a PhotoBook document:

```
tell application "HyperCard"
   copy picture of current background to BkgndPaint
end tell

tell application "PhotoBook"
   activate
   set clipboard to BkgndPaint
   paste
end tell
```

picture XCMD

The Picture XCMD can now accept PICT data as a parameter, as well as a filename, the clipboard (when it contains PICT data), or the name of a PICT resource. Pass the string `appleEvent` as the second parameter and pass the picture data of type `'PICT'` in any later parameter, as the following AppleScript example shows.

```
on mouseUp
  tell application "Microsoft Excel"
    make Document
    copy "=sin(row()/8)*cos(column()/4)" to ¬
    first Cell of first Row of first Document
    copy first Cell to Range "R1C1:R20C10"
    make Chart
    copy {10,60,600,400} to bounds of first Window
    copy three D Surface to type of first Chart
    copy first Chart to theChart
  end tell
  |picture|("Excel chart", "appleEvent", theChart)
end mouseUp
```

You can use the Picture XCMD within an appleEvent handler to display the PICT image contained in the direct parameter of an Apple event. Use the following form:

```
on appleEvent class,id,sender
  if class & id is "some custom event class and id"
   picture "name","appleEvent",...
  else pass appleEvent
end appleEvent
```

This works for events that contain a picture in their direct parameter, but not for events that contain aliases to PICT files such as an `'odoc'` event.

text ranges

HyperCard supports negative indices for text ranges in OSA scripting languages.

Negative indices are interpreted as offsets from the end of the text. So `character -1 of card field 1` in AppleScript is the last character of card field 1.

Appendix C
Quick Color Tutorial

This appendix teaches the bare essentials of using color in a stack. After you've taken this tutorial, you'll know how to

- open the color editor
- use several icons on the color palette
- add color to buttons and fields
- create a color backdrop
- move colors to their proper layers

What you should already know: Before you take this tutorial, you should have read the Preface to this guide, and you should understand the elements of HyperCard described in Chapter 1 of the *HyperCard Reference Manual*.

Making sure Color Tools is installed

Before you begin, make sure that Color Tools are already installed in your Home stack, and that they are active.

- **Look in the menu bar for a menu named Colors.**

  If you see this menu, Color Tools are installed. Go on to the next section, "Preparing to Add Color."

- **Look on the first card of the Home stack for an icon labeled "Color Tools are OFF" and click it.**

Color Tools are OFF

The icon's label changes to "Color tools are ON," and you see the Colors menu in the menu bar.

Color Tools are ON

Go to the next section, "Preparing to Add Color."

- **Install the Color Tools by following the instructions in Chapter 3, "Installing Color Tools."**

  When the tools are installed, continue with this tutorial.

Preparing to add color

In this tutorial you'll be using a copy of the Practice stack. You'll work with the copy so that the original will be preserved in case you want to start over.

To create a copy of the stack and to open it, follow these steps:

1 **Click the Practice icon on the first card of the Home stack.**

The Practice stack opens.

2 **Choose Save a Copy from the File menu.**

A dialog box appears.

3 **Type "Color Practice" into the name field and click Save.**

4 **Open the Color Practice stack.**

Choose Open Stack from the File menu and locate the Color Practice stack. Then click Open.

The Practice stack closes, and the Color Practice stack opens.

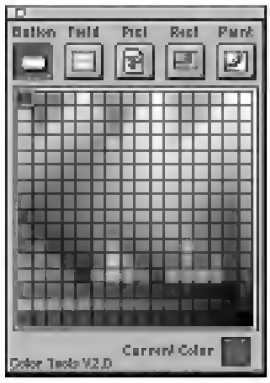
Opening the color editor

When you work in color, you leave the standard HyperCard environment and enter the color editor.

- **To open the color editor, choose Open Coloring Tools from the Color menu.**

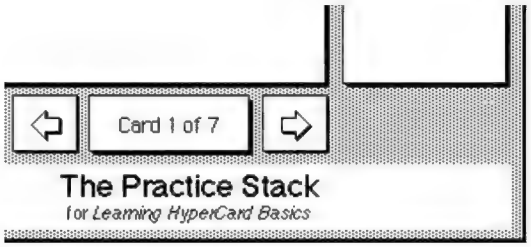
  A dialog box appears, asking you to confirm that you want to add color resources to the stack. Click OK.

Any palettes open in HyperCard disappear, and the color editor opens. The color palette appears on the screen.



Adding color to buttons

In this exercise you'll add color to the Previous Card and Next Card buttons.



1 **Click the Button icon at the upper-left corner of the color palette.**

2 **Click the Previous Card button to select it.**

3 **Click a light color on the color palette to select it.**

The button takes on the color you choose.

A dark color may obscure the design of the button. If the color you choose is too dark, click another color.

4 **Click the Next Card button to select it.**

5 **Click the same color on the palette as you did for the Previous Card button.**

The color has a small box around it.

Both buttons now have the same color.

Adding a color backdrop

In this exercise you'll add a color backdrop to the background of the stack.

1. **Choose Background from the Edit menu to open the background.**
2. **Double-click the Rect icon at the top of the color palette.**

   A small rectangle about 1/2-inch square appears on the screen in the current color.

3. **Click a light color on the color palette.**

   Select a color different from the one you used for the buttons, but make sure that it's light.

4. **Drag the rectangle by its center to the upper-left corner of the stack.**

   You can tell that you're dragging from the right place if the pointer is replaced by a hand.

5. **Size the rectangle so that it fills the entire background.**

   To size the rectangle, grab it by its lower-right corner and drag diagonally to the lower-right corner of the stack.

The color of the backdrop you just created obscures the color of the buttons.

6 **Choose Send to Back from the Items menu.**

The backdrop moves behind the buttons, and the button color shows over the backdrop.

7 **Choose Background from the Edit menu to close the background.**

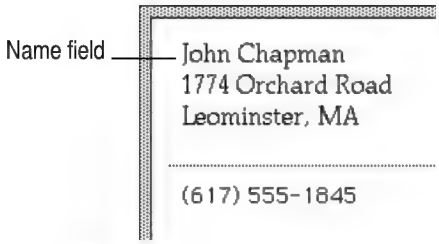
The text on the card level becomes visible again.

Adding color to fields

Now you'll add color to the fields.

1 **Click the Field icon at the top of the color palette.**

2 **Click the field that has the name in it to select it.**



3 **Click a different light color on the color palette.**

It's especially important to choose a light color for a field so that the color doesn't obscure the field's text.

If you don't like the color that appears on the field, click another color.

4 **If you want, add color to the other fields on the card.**

Closing the color editor

Now you'll close the color editor and check how the color looks as you change cards.

1. **Click the close box on the color palette to close the color editor.**

   You return to the standard HyperCard environment.

2. **Click the Next Card button.**

   The color remains as you change cards. All the buttons and fields you colorized are in the background, so the color remains visible.

   If you decide you want to change any of the colors you've used, reenter the color editor, select the proper icon on the palette, click the element whose color you want to change, and then click a new color.

   You can also return to the color editor and add color to other buttons or fields in the stack, or add smaller colorized rectangles to highlight different parts of the stack.

Where to go from here

To learn more about the color editor, see Chapter 4, "Understanding the Color Editor."

To learn how to create color pictures, see Chapter 7, "Working with Color Paint Tools."

Or you can go back into the color editor and experiment on your own with the Color Practice stack.

Index

A

`AC RemoteInstall` scripting command 85
`addButton` scripting command 86
`addColor` external scripting command (XCMD)
  general structure and usage 84
  installing under script control 85
`addField` scripting command 87
`addPictFile` scripting command 89
`addPict` scripting command 88
`addRect` scripting command 90
aliases for PICT files 50
Anti–Alias (Effects menu) 67
Apple events, updated/corrected documentation notes 144–146
Apple HSL (Hue–Saturation–Lightness) color number system 31–32
Apple RGB (Red–Green–Blue) color number system 32–33

B

background color, adding 58, 67
Background (Edit menu) 22, 152
Background pop–up menu 58
Background Transitions (Effects menu) 25
Bevel pop–up menu 39
bevels, adding to an object 39–40
bit depth of Color Tools, changing 52
borderless shapes, drawing 65
brightness, in HSB color number system 32
Bring Closer (Items menu) 22
Bring to Front (Items menu) 22
brush tool 61
built–in tasks, assigning 4–5
Button icon on color palette 28, 151
Button Info dialog box 141
Button Info (Objects menu) 2
buttons
  adding color to 151
  assigning tasks to 3–6
  assigning visual effects to 6
  creating 2
  finding borderless uncolored buttons 36
  setting to go to a card 4–5
Button Tasks window, using 1–6
Button tool 2

C

card, setting a button to go to 4–5
Card Transitions (Effects menu) 25
`changeObjectBevel` scripting command 91
`changeObjectBounds` scripting command 92
`changeObjectColor` scripting command 93
`changeObjectTransparency` scripting command 94
closed shapes, creating 65–66
closing
  the color editor 20, 139, 154
  the PICT editor 76
color
  adding to an object 36–40, 57–58
  adding to buttons 151
  adding to fields 153
  adjusting in the Info box 38
  changing to opposite side of the color spectrum 67
  doesn't show, troubleshooting 138
  editing 72–73
  preparing a stack for 16
  selecting from part of an image 64
  visual effects and 6
color backdrop, adding to a stack 152–153
`colorBackground` scripting command 95
`colorButton` scripting command 96
`colorCardLayered` scripting command 99–100
`colorCard` scripting command 97–98
color editor
  can't select an item, troubleshooting 138
  closing 20, 139, 154
  color number systems 28–33
  finding borderless uncolored buttons/fields 36
  opening 16, 20, 150
  tutorial 147–154
  using the color palette 27–28
  using the menus 21–26
`colorField` scripting command 101
coloring existing stacks 80
colorized cards, maximizing speed for 78–79
colorized stacks, troubleshooting 137–138
colorizing buttons and fields 35–41
color layers, defined 20–21
Color menu
  Open Coloring Tools 16, 20, 26, 150
  Redraw Screen 26, 139
color number systems
  Apple HSL (Hue–Saturation–Lightness) system 31–32
  Apple RGB (Red–Green–Blue) system 32–33
  HSB (Hue–Saturation–Brightness) system 30–31
  overview 28
  RGB (Red–Green–Blue) system 29–30
color operations, maximizing speed 78–79
color overlays
  adding to objects 36–40
  choosing color for 28–33
  defined 20–21
  editing 23, 41

color paint tools palette
brush tool 61
closing the PICT editor from 76
color pick–up tool 64
eraser tool 64
icons 55, 57
icons, second group for shortcuts 71
lasso tool 60
line tool 65
marquee tool 59
paint bucket tool 62
pencil tool 61
spray can tool 62
text tool 63
using 57–59
color palette, using 27–28
color picker
opening 24, 38
using sliders to set RGB values 32–33
color pick–up tool 64
`colorPictFile` scripting command 104–105
`colorPict` scripting command 102–103
`colorRect` scripting command 106–107
color stacks, using efficiently 77–80
color standalone application, saving a stack as 16–17
Color Tools
changing bit depth 52
installing into your Home stack 13–17, 148
turning on or off 15, 137, 138, 148
Color Tools button 14
Color Tools palette 37
Color Tools stack icon 14
color tutorial 147–154
Compact Color Database (Edit menu) 22
`compact` scripting command 108–109
Compact Stack (File menu) 16
coordinates, setting 40
Copy (Edit menu) 41
copying a stack 16, 149
corrected/updated documentation notes 141–146
Create New Picture (Items menu) 22, 56
cropping PICT images 51
curved shapes, drawing 66
Cut (Edit menu) 22, 41

D

dialog boxes
Button Info 141
Import Picture 47
Item Info 23
Select a Picture 45, 48
`disableObject` scripting command 111
`disable` scripting command 110
disk space, conserving when working with color stacks 78
displaying
PICT files 49–50
PICT resources 48–49
dithering 51–52
documentation
how to use xiii–xiv
updated/corrected information 141–146
Draw Centered (Options menu) 65, 70
Draw Filled (Options menu) 65, 70
drawing
centered on a given point 65, 70
closed shapes 65–66
free–form lines 61
multiple images 70
Draw Multiple (Options menu) 65, 70

E

edges
  outlining 68
  smoothing 67
Edit Colors (Options menu) 72–73
Edit Gradient (Options menu) 74–75
editing
  color overlays 23, 41
  colors in the PICT editor 72–73
  patterns 58–59
  PICT images 54–55
Edit menu
  Background 152
  command summary 22
  Copy 41
  Cut 41
  Paste 41
  Undo 41
Edit Picture (Items menu) 22, 55
Effects menu 25, 67–69
`enableObject` scripting command 113
`enable` scripting command 112
eraser tool 64
erasing an image 64
Export Graphics (File menu) 72
exporting an image from the PICT editor 72

F

Field icon on color palette 28
fields
  adding color to 153
  finding borderless uncolored fields 36
File menu
  Compact Stack 16
  Export Graphics 72
  Get Info 17, 50
  Import Graphics 71
  Quit 76
  Quit HyperCard 21
  Revert 72
  Save 72
  Save a Copy 16, 149
Fill (Effects menu) 67
filling
  with the current pattern and colors 67
  solid areas of an image 62
Flip (Effects menu) 69
flipping an image 69
foreground color, adding 57, 67
Foreground pop–up menu 57
free–form lines, drawing 61

G

`getBitsCall` scripting command 114
`getButtonIndex` scripting command 115–116
`getFieldIndex` scripting command 117
Get Info (File menu) 17, 50
`getObjectBevel` scripting command 119
`getObjectBounds` scripting command 120
`getObjectClicked` scripting command 121–122
`getObjectColor` scripting command 123
`getObjectType` scripting command 124
`getPictName` scripting command 125
gradient, creating 74–75

H

height of an image, changing. *See* scaling PICT images
Home stack, installing Color Tools into 13–17, 148
HSB (Hue–Saturation–Brightness) color number system 30–31
hue, in HSB color number system 31, 32
HyperTalk
  new commands for HyperCard 2.3 7–11
  updated/corrected documentation notes 142–144

I

icons
  Button 151
  Color Tools stack 14
  on color paint tools palette 55, 57, 71
  on color palette 28
  Pict 28, 45, 54
Import Graphics (File menu) 71
importing
  an image into the PICT editor 71
  a PICT file from disk 45–46
  a PICT image from another stack 46–47
  PICT images, troubleshooting 138
Import Picture dialog box 47
Info box
  changing an object's position 40
  changing an object's RGB numbers 39
  changing an object's size 40
  scaling a PICT image 50–51
  using a colorized object's Info box 38
Installer 1 disk xii, xiii
installing
  Color Tools 13–17, 148
  HyperCard 2.3 xii–xiii
`install` scripting command 126
intensities of colors in RGB color number system 29–30
Invert (Effects menu) 67
Item Info dialog box 23
Item Info (Items menu) 22, 32, 38, 138
Items menu
  command summary 22
  Create New Picture 22, 56
  Edit Picture 22, 55
  Item Info 22, 32, 38, 138
  Place Picture 22, 45, 46, 48, 49
  Send to Back 22, 153

J, K

jagged edges, smoothing 67

L

lasso tool 60
layers, defined 20–21
line tool 65

M

MacPaint images,
importing/exporting 71–72
magnification, setting 73
marquee tool 51, 59
memory, optimizing for color
stacks 77, 138
memory requirements
for Color Tools 13
for HyperCard 2.3 xii
for standalone applications 17, 138
`moveBackward` scripting
command 127–128
`moveForward` scripting command 129
`moveToBack` scripting command 130
`moveToFront` scripting command 131
multiple shapes, drawing 65

N

New Button (Objects menu) 2
number systems for colors. *See* color
number systems

O

Objects menu
Button Info 2
New Button 2
Opaque (Effects menu) 69
Open Coloring Tools
(Color menu) 16, 20, 26, 150
opening
color editor 16, 20, 150
color picker 24, 38
PICT editor 54
operating system requirements. *See*
system requirements
Options menu
Draw Centered 65
Draw Filled 65
Draw Multiple 65
Edit Colors 72–73
Edit Gradient 74–75
Zoom In/Out 73
outlining edges 68
oval shapes, drawing 66
overlays. *See* color overlays

P

paint bucket tool 62
paint effects
setting 67–70
shortcuts for 71
Paint icon on color palette 28
painting with the current pattern and
brush 61
palette. *See* color paint tools palette; color
palette; Color Tools palette;
system palette
Paste (Edit menu) 41
patterns, choosing and editing 58–59
Patterns pop–up menu 57, 58–59
pencil tool 61
PICT editor
closing 76
creating a gradient 74–75
editing colors 72–73
importing/exporting images 71–72
opening 54
saving images 72
setting paint effects 67–71
zooming 73
PICT files
adding from disk 45–46
aliases for 50
copying from another stack 46–47
displaying on disk 49–50
vs. PICT resources 44
Pict icon on color palette 28, 45, 54

PICT images
adding to a stack as a resource 45–47
applying effects to 67–71
creating 56
cropping 51
dithering 51–52
editing 54–55
importing/exporting 71–72
saving 72
scaling 50–51, 69
using the color paint tools palette 57–59
PICT resources
displaying in a stack 48–49
vs. PICT files on disk 44
`Picture` HyperTalk command 52
Place Picture (Items menu) 22, 45, 46, 48, 49
Place Rectangle (Items menu) 22
polygons with irregular sides, creating 66
polygon tool 66
position of an object, changing 38, 40
proportions of an image, changing. *See* scaling PICT images

Q

Quit HyperCard (File menu) 21
quitting the PICT editor 76

R

Read Me file xii, xiii
rectangles, drawing 66
Rect icon on color palette 28
Redraw Screen (Color menu) 26, 139
`remove` scripting command 132
`removeButton` scripting command 133
`removeField` scripting command 134
`removeObject` scripting command 135
restoring a scaled image to original size and proportions 51
Revert (File menu) 72
RGB (Red–Green–Blue) color number system 29–30
changing numbers for an object 39
Rotate (Effects menu) 68
rotating an image 68

S

saturation, in HSB color number system 31, 32
Save a Copy (File menu) 16, 149
Save (File menu) 72
saving PICT images 72
Scale (Effects menu) 69
scaling PICT images 50–51, 69
scripting functions
`AC RemoteInstall` command 85
`addButton` command 86
`addColor` XCMD 84, 85
`addField` command 87
`addPict` command 88
`addPictFile` command 89
`addRect` command 90
`changeObjectBevel` command 91
`changeObjectBounds` command 92
`changeObjectColor` command 93
`changeObjectTransparency` command 94
`colorBackground` command 95
`colorButton` command 96
`colorCard` command 97–98
`colorCardLayered` command 99–100
`colorField` command 101
`colorPict` command 102–103
`colorPictFile` command 104–105
`colorRect` command 106–107
`compact` command 108–109
`disable` command 110
`disableObject` command 111
`enable` command 112

enableObject command 113
getBitsCall command 114
getButtonIndex command 115–116
getFieldIndex command 117–118
getObjectBevel command 119
getObjectBounds command 120
getObjectClicked command 121–122
getObjectColor command 123
getObjectType command 124
getPictName command 125
install command 126
moveBackward command 127–128
moveForward command 129
moveToBack command 130
moveToFront command 131
remove command 132
removeButton command 133
removeField command 134
removeObject command 135
sort command 136
syntax terms and conventions 82–83
scripts, adding with the Button Tasks window 1–6
Select a Picture dialog box 45, 48
selecting
a color from part of an image 64
different parts of an image 59, 60
an image between other graphics 60
Send Farther (Items menu) 22
Send to Back (Items menu) 22, 153
shutting down the computer 139
size of an object, changing 38, 40
software requirements. *See* system requirements
Sort Color Database (Edit menu) 22
sort scripting command 136
sound, system requirements for 10
speak HyperTalk command 8
speech, system requirements for xii, 8
speed, maximizing for colorized cards 78–79
speed settings for transition effects 25
spray can tool 62
spraying color onto an area 62
squares, drawing 66
stacks
adding a color backdrop to 152–153
adding a PICT image to 45–47
changing the openStack handler 137
coloring an existing stack 80
displaying a PICT resource in 48–49
preparing for color 16
using a PICT file in 49–50
Stack Transitions (Effects menu) 25
standalone applications
changing memory allocation 17
color doesn't show, troubleshooting 137, 138
saving stacks as 16–17
stop sound HyperTalk command 11
stop speech HyperTalk command 9
straight lines, drawing 65
system palette 44, 51
system requirements
for all features xii
for sound 10
for speech 8

T

tasks, assigning using the Button Tasks window 3–6
text, adding to an image 63
Text menu 63
text tool 63
`the clipboard` HyperTalk command 11
`the soundChannel` HyperTalk command 10
`the speech` HyperTalk command 9
`the voices` HyperTalk command 9
3–D effects
  adding shading with the Bevel pop–up menu 39
  setting/changing in the Info box 38
Tint (Effects menu) 67
tinting an image 67
Trace Edges (Effects menu) 68
transition effects, setting 25, 26
Transparent (Effects menu) 69
troubleshooting
  can't select an item in the color editor 138
  can't shut down the computer 139
  color doesn't follow when you move a colorized object 139
  color doesn't show 137, 138, 139
  coloring existing stacks 80
  Color menu doesn't appear in menu bar 137
  HyperCard visual effects stopped working 138–139
  Item Info isn't available for an object 138
  monitor doesn't display enough colors 137
  PICT image doesn't appear in stack 138
  removing color from a colorized stack 138
turning an image 68
turning Color Tools on/off 15, 137, 138
tutorial 147–154

U

Undo (Edit menu) 22, 41
updated/corrected documentation notes 141–146

V

visual effects
  assigning to buttons 6
  color and 6, 26
  preventing standard visual effects from working as you change cards 26
  stopped working, troubleshooting 138–139

W, X, Y

width of an image, changing. *See* scaling PICT images

Z

Zoom box on color paint tools palette 71
zooming an image 73
Zoom In/Out (Options menu) 73

Getting Started

With HyperCard 2.3

Apple Computer, Inc.

© 1995 Apple Computer, Inc. All rights reserved.

Under the copyright laws, this manual may not be copied, in whole or in part, without the written consent of Apple. Your rights to the software are governed by the accompanying software license agreement.

The Apple logo is a trademark of Apple Computer, Inc., registered in the U.S. and other countries. Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for printing or clerical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014-2084
(408) 996-1010

Apple, the Apple logo, HyperCard, HyperTalk, Macintosh, QuickTime, and PlainTalk are trademarks of Apple Computer, Inc., registered in the U.S. and other countries. AppleScript, Mac, MacinTalk, Power Macintosh, and ResEdit are trademarks of Apple Computer, Inc.

Classic is a registered trademark licensed to Apple Computer, Inc.

PhotoDisc™ ClipPix™ Images © 1994 PhotoDisc, Inc.

AddMotion is a trademark of Motion Works International.

Adobe, Adobe Illustrator, Adobe Photoshop, and PostScript are trademarks of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

ExposurePro is a registered trademark of Baseline Publishing, Inc.

Helvetica and Times are registered trademarks of Linotype-Hell AG and/or its subsidiaries.

Microsoft is a registered trademark of Microsoft Corporation.

QMS is a registered trademark of QMS, Inc.

QuarkXPress is a registered trademark of Quark, Inc.

SuperPaint is a registered trademark of Aldus Corporation, a subsidiary of Adobe Systems Incorporated, which may be registered in certain jurisdictions.

Tektronix is a registered trademark of Tektronix, Inc.

Trinitron is a trademark of Sony Corporation, registered in the U.S. and other countries.

Simultaneously published in the United States and Canada.

Mention of third-party products is for informational purposes only and constitutes neither an endorsement nor a recommendation. Apple assumes no responsibility with regard to the performance or use of these products.

Contents

Preface Welcome xi

System requirements xii

Installing HyperCard 2.3 xii

What you must already know xiii

Using HyperCard documentation xiii

Finding information in HyperCard documentation xiii

What's in this guide xv

1 Using the Button Tasks Window 1

Opening the Button Tasks window 2

Choosing tasks 3

Assigning built-in tasks 4

Going to a card 4

Adding a visual effect 6

2 Using the New HyperTalk Commands 7

Text to Speech 8

speak 8

the speech 9

stop speech 9

the voices 9

Sound 10

the soundChannel 10

stop sound 11

Clipboard 11

the clipboard 11

3 Installing Color Tools 13

Installing color tools into your Home stack 13

Turning Color Tools on and off 15

Preparing a stack for color 16

Preparing a color standalone application 16

4 Understanding the Color Editor 19

Understanding overlays and layers 20

Using color editor menus 21

Using the color palette 27

Changing an element's color 27

Color palette icons 28

Coloring by the numbers 28

RGB (Red-Green-Blue) system 29

HSB (Mac OS 7.1) 30

Apple HSL (Mac OS 7.5) 31

Apple RGB (Mac OS 7.5) 32

5 Colorizing Buttons and Fields 35

Adding color to an object 36

Using the Info box 38

Adjusting RGB numbers 39

Adding 3-D effects 39

Changing an object's position 40

Changing an object's size 40

Editing a colorized object's overlay 41

Using menu commands 41

Copying a color from one object to another 41

6 Using Pictures 43

PICT resources versus PICT files on disk 44

PICT resources 44

PICT files on disk 44

Adding a PICT image to your stack as a resource 45

Adding a PICT file from disk 45

Copying a PICT image from another stack 46

Displaying a PICT resource 48

Displaying a PICT file on disk 49

Scaling PICT images 50

Cropping PICT images 51

Dithering 51

7 Working with Color Paint Tools 53

Entering the PICT editor 54

Preparing to edit an existing PICT image 54

Preparing to create a new PICT image 56

Using the color paint tools palette 57

Choosing and editing patterns 58

Using the color paint tools 59
Marquee tool 59
Lasso tool 60
Pencil tool 61
Brush tool 61
Paint bucket tool 62
Spray can tool 62
Text tool 63
Eraser tool 64
Color pick-up tool 64
Line tool 65
Creating closed shapes 65
Polygon tool 66
Setting effects 67
Fill 67
Invert 67
Tint 67
Anti-Alias 67
Trace Edges 68
Rotate 68
Scale 68
Flip 69
Opaque 69
Transparent 69
Draw Filled 70
Draw Centered 70
Draw Multiple 70
Paint effect shortcuts 71

Importing and exporting images 71
Importing an image 71
Exporting an image 72
Saving a PICT image 72
Special features of the PICT editor 72
Editing colors 72
Zooming 73
Creating a gradient 74
Closing the PICT editor 76

8 Maximizing color stacks 77

Memory 77
Disk space 78
Speed 78
Bit depth 78
Changing a card 78
Moving from card to card 79
Types of object 79
PICT size 79
Number of objects 79
Coloring existing stacks 80
Troubleshooting 80

9 Scripting for Color 81

Syntax terms and conventions used in this chapter 82

The addColor XCMD 84

AddColor anatomy 84

AC_RemoteInstall 85

addButton 86

addField 87

addPict 88

addPictFile 89

addRect 90

changeObjectBevel 91

changeObjectBounds 92

changeObjectColor 93

changeObjectTransparency 94

colorBackground 95

colorButton 96

colorCard 97

colorCardLayered 99

colorField 101

colorPict 102

colorPictFile 104

colorRect 106

compact 108

disable 110

disableObject 111

enable 112

enableObject 113

getBitsCall 114

getButtonIndex 115

getFieldIndex 117
getObjectBevel 119
getObjectBounds 120
getObjectClicked 121
getObjectColor 123
getObjectType 124
getPictName 125
install 126
moveBackward 127
moveForward 129
moveToBack 130
moveToFront 131
remove 132
removeButton 133
removeField 134
removeObject 135
sort 136

Appendix A Troubleshooting 137

Appendix B Updates 141

HyperCard 141
HyperTalk 142
Apple events 144

Appendix C Quick Color Tutorial 147

Making sure Color Tools are installed 148

Preparing to add color 149

Opening the color editor 150

Adding color to buttons 151

Adding a color backdrop 152

Adding color to fields 153

Closing the color editor 154

Where to go from here 154

Index 155

Welcome

Congratulations on purchasing HyperCard version 2.3, the software construction set for the rest of us. In addition to everything included in previous versions, HyperCard 2.3 includes

- full speed compatibility for all Power Macintosh computers
- autoscripting Button Tasks, including multimedia and navigation support at the click of a button
- script commands that allow you to include speech and multichannel sound in your stacks
- color painting tools to create new PICT images or to edit and add color to existing PICT images

IMPORTANT Read this chapter before you read any other documentation that comes with HyperCard. It supersedes all other information about system requirements and software installation, and about how to use the extensive documentation. Appendix B, "Updates," corrects information that has changed since the rest of the documentation was printed and provides new information about features that have been improved since the last version of HyperCard was released.

System requirements

HyperCard 2.3 works with Macintosh operating system (Mac OS) versions 6.0.5 and later.

To use Apple Open Scripting Architecture (OSA), you'll need Mac OS 7.0 or higher and the AppleScript extension.

To use AppleScript, you'll need Mac OS 7.0 or higher and AppleScript 1.1 or higher.

To use video, you'll need QuickTime 1.6.1 or Mac OS 7.5 or higher.

To use text to speech conversion, you'll need the Speech Manager extension, MacinTalk Pro extension, and Voices folder from the PlainTalk software package. (If you are operating HyperCard under Mac OS 7.5 with a Macintosh AV computer or with a Power Macintosh computer, use PlainTalk version 1.3. Otherwise use PlainTalk 1.2.1.)

To use Button Tasks, you'll need Mac OS 7.1 or Mac OS 7.0 plus either QuickTime or AppleScript 1.1 or higher.

HyperCard 2.3 requires a minimum of 1050K of random access memory (RAM) for its own use (2200K if you use color). To use HyperCard with AppleScript, you'll need to set HyperCard's minimum memory to 2200K.

HyperCard for Power Macintosh needs 3072K (5120K for color). For best results, turn virtual memory off in the Memory control panel.

Installing HyperCard 2.3

To install your new software, follow these instructions:

1. **Insert the disk labeled *Installer 1* into the disk drive.**
2. **If the disk icon is not already open, double-click to open it.**
3. **Double-click the Read Me file to open it; then read its contents.**

   If the installation instructions in the Read Me file differ from what you see here, follow the instructions in the Read Me file.
4. **Click Install.**

The version of HyperCard appropriate to your computer is automatically installed.

5 **Follow the directions that appear on the screen.**

6 **To install color tools in your Home stack or to prepare a stack for color, follow the instructions in Chapter 3 of this guide.**

7 **To install additional programs packaged with this version of HyperCard, see the documentation that came with the programs.**

Be sure to check the product's Read Me file as well as the Read Me file on the HyperCard *Installer 1* disk for any last minute changes.

What you must already know

To understand and use the HyperCard material in this guide, you should already understand the elements of HyperCard described in Chapter 1 of the *HyperCard Reference Manual.*

To understand and use the material on scripting, you should already know how to write scripts. See the *HyperCard Script Language Guide* and the HyperTalk Reference stack for information.

Using HyperCard documentation

HyperCard comes with a complete documentation package. Its various parts (except for this guide) are described on page xviii of the *HyperCard Reference Manual.* This section describes which parts you should use to accomplish specific tasks.

Finding information in HyperCard documentation

Information sometimes varies slightly in HyperCard documentation. In the following list, information in a lower numbered document takes precedence over information in a higher numbered document.

1. "Read Me First" file on the first HyperCard installation disk
2. This guide
3. New Features stack

4. Help stacks
5. Quick Reference Card
6. *HyperCard Reference Manual* and *HyperCard Script Language Guide*

If you find that information in a printed document (except for this guide) is in conflict with information in a Help stack, rely on the stack.

Learning HyperCard

- For an overview of most of HyperCard's features, see Chapter 1 of the *HyperCard Reference Manual.*
- For a tutorial that teaches HyperCard's key features, see Chapter 2 of the *HyperCard Reference Manual.*

Learning about color

- For an overview of HyperCard's color features, see Chapter 4 of this guide.
- For detailed reference material about color, see chapters 3 through 9 of this guide.
- For a tutorial on the basics of HyperCard's color features, see Appendix C of this guide. (Review Chapter 3 of this guide first.)

Learning the new features

- For information on all the features (except color) added to HyperCard since version 2.1, see the New Features Stack.

Learning HyperTalk

- For an overview of HyperTalk, the scripting language of the HyperCard environment, see Chapter 2 of the *HyperCard Script Language Guide.*
- For a tutorial that teaches the basic elements of scripting in HyperTalk, see Chapter 3 of the *HyperCard Reference Manual.*

What's in this guide

This guide describes the new features in HyperCard 2.3, and includes complete documentation on the HyperCard color tools.

- Chapter 1, "Using the Button Tasks Window," describes how to add features to your stacks without scripting.
- Chapter 2, "Using the New HyperTalk Commands," gives complete details on the HyperTalk commands new in version 2.3, including commands for speech and multichannel sound.
- Chapter 3, "Installing Color Tools," describes how to prepare your Home stack so that you can use the color tools, how to turn the color tools on and off, and how to add color tools to a specific stack.
- Chapter 4, "Understanding the Color Editor," presents an overview of the editor, describing briefly what you can do with it and how to use it.
- Chapter 5, "Colorizing Buttons and Fields," describes how to add overlays to buttons and fields, how to add 3-D effects to a colorized object, how to modify a color with RGB numbers, how to change an object's size or position, and how to modify an object's color overlay.
- Chapter 6, "Using Pictures," describes how to display high-resolution PICT files and resources. It also covers the costs and benefits of storing PICT images directly in a stack as opposed to on disk.
- Chapter 7, "Working with Color Paint Tools," explains how to use the color paint tools to create and edit color PICT images, and how to modify PICT images that you import.
- Chapter 8, "Maximizing Color Stacks," provides tips for getting the most out of color stacks. Topics include how to speed stacks up and how to make more efficient use of memory and disk space.
- Chapter 9, "Scripting for Color," describes the scripting functions that the color tools add to HyperTalk, HyperCard's scripting language.
- Appendix A "Troubleshooting," covers some common areas of confusion and what to do about them.
- Appendix B, "Updates," corrects information that has changed since the rest of the documentation was printed and describes features that have been improved.
- Appendix C, "Quick Color Tutorial," teaches the basics of HyperCard's color features.